# Avid®

# iNEWS™ Newsroom Computer System

## Operations and Troubleshooting Manual

Version 2.0

make manage move | media™ Avid®

## Copyright and Disclaimer

Product specifications are subject to change without notice and do not represent a commitment on the part of Avid Technology, Inc.

The software described in this document is furnished under a license agreement. You can obtain a copy of that license by visiting Avid's Web site at www.avid.com. The terms of that license are also available in the product in the same directory as the software. The software may not be reverse assembled and may be used or copied only in accordance with the terms of the license agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

Avid products or portions thereof are protected by one or more of the following United States patents: 4,746,994; 4,970,663; 5,045,940; 5,267,351; 5,309,528; 5,355,450; 5,396,594; 5,440,348; 5,452,378; 5,467,288; 5,513,375; 5,528,310; 5,557,423; 5,568,275; 5,577,190; 5,584,006; 5,640,601; 5,644,364; 5,654,737; 5,715,018; 5,724,605; 5,726,717; 5,729,673; 5,745,637; 5,752,029; 5,754,851; 5,781,188; 5,799,150; 5,812,216; 5,852,435; 5,905,841; 5,929,836; 5,929,942; 5,930,445; 5,930,797; 5,946,445; 5,987,501; 5,999,190; 6,016,152; 6,018,337; 6,023,531; 6,023,703; 6,031,529; 6,058,236; 6,061,758; 6,091,778; 6,105,083; 6,118,444; 6,128,001; 6,134,607; 6,137,919; 6,141,691; 6,157,929; 6,160,548; 6,167,404; 6,198,477; 6,201,531; 6,223,211; 6,249,280; 6,269,195; 6,317,158; 6,317,515; 6,330,369; 6,351,557; 6,353,862; 6,357,047; 6,392,710; 6,404,435; 6,407,775; 6,417,891; 6,426,778; 6,449,019; 6,473,094; 6,477,271; 6,489,969; 6,512,522; 6,532,043; 6,539,163; D396,853; D398,912. Additional U.S. and foreign patents pending.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Avid Technology, Inc.

Copyright © 2004 Avid Technology, Inc. and its licensors. All rights reserved. Printed in USA.

The following disclaimer is required by Sam Leffler and Silicon Graphics, Inc. for the use of their TIFF library:

Copyright © 1988–1997 Sam Leffler
Copyright © 1991–1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software [i.e., the TIFF library] and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED “AS-IS” AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

### The following disclaimer is required by the Independent JPEG Group:

Portions of this software are based on work of the Independent JPEG Group.

### The following disclaimer is required by Paradigm Matrix:

Portions of this software licensed from Paradigm Matrix.

### The following disclaimer is required by Ray Sauers Associates, Inc.:

“Install-It” is licensed from Ray Sauers Associates, Inc. End-User is prohibited from taking any action to derive a source code equivalent of “Install-It,” including by reverse assembly or reverse compilation, Ray Sauers Associates, Inc. shall in no event be liable for any damages resulting from reseller’s failure to perform reseller’s obligation; or any damages arising from use or operation of reseller’s products or the software; or any other damages, including but not limited to, incidental, direct, indirect, special or consequential Damages including lost profits, or damages resulting from loss of use or inability to use reseller’s products or the software for any reason including copyright or patent infringement, or lost data,

even if Ray Sauers Associates has been advised, knew or should have known of the possibility of such damages.

The following disclaimer is required by Videomedia, Inc.:

"Videomedia, Inc. makes no warranties whatsoever, either express or implied, regarding this product, including warranties with respect to its merchantability or its fitness for any particular purpose."

"This software contains V-LAN ver. 3.0 Command Protocols which communicate with V-LAN ver. 3.0 products developed by Videomedia, Inc. and V-LAN ver. 3.0 compatible products developed by third parties under license from Videomedia, Inc. Use of this software will allow "frame accurate" editing control of applicable videotape recorder decks, videodisc recorders/players and the like."

The following disclaimer is required by Altura Software, Inc. for the use of its Mac2Win software and Sample Source Code:

©1993–1998 Altura Software, Inc.

Attn. Government User(s). Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software and its documentation are "commercial computer software" or "commercial computer software documentation." In the event that such Software or documentation is acquired by or on behalf of a unit or agency of the U.S. Government, all rights with respect to this Software and documentation are subject to the terms of the License Agreement, pursuant to FAR §12.212(a) and/or DFARS §227.7202-1(a), as applicable.

Trademarks

888 I/O, AirPlay, AirSPACE, AirSPACE HD, AniMatte, AudioSuite, AudioVision, AutoSync, Avid, AVIDdrive, AVIDdrive Towers, AvidNet, AvidNetwork, AVIDstripe, Avid Unity, Avid Xpress, AVoption, AVX, CamCutter, ChromaCurve, ChromaWheel, DAE, D-Fi, D-fx, Digidesign, Digidesign Audio Engine, Digidesign Intelligent Noise Reduction, DigiDrive, DigiTranslator, DINR, D-Verb, Equinox, ExpertRender, FieldPak, Film Composer, FilmScribe, FluidMotion, HIIP, HyperSPACE, HyperSPACE HDCAM, IllusionFX, Image Independence, Intraframe, iS9, iS18, iS23, iS36, Lo-Fi, Magic Mask, make manage move | media, Marquee, Matador, Maxim, MCXpress, Media Composer, MediaDock, MediaDock Shuttle, Media Fusion, Media Illusion, MediaLog, Media Reader, Media Recorder, MEDIArray, MediaShare, Meridien, MetaSync, NaturalMatch, Nearchive, NetReview, NewsCutter, OMF, OMF Interchange, OMM, Open Media Framework, Open Media Management, ProEncode, Pro Tools, QuietDrive, Recti-Fi, RetroLoop, rS9, rS18, Sci-Fi, Softimage, Sound Designer II, SPACE, SPACEShift, Symphony, the Avid|DS logo, Trilligent, UnityRAID, Vari-Fi, Video Slave Driver, VideoSPACE, and Xdeck are either registered trademarks or trademarks of Avid Technology, Inc. in the United States and/or other countries.

iNEWS, iNEWS ControlAir, and Media Browse are trademarks of iNews, LLC.

All other trademarks contained herein are the property of their respective owners.

Footage

Arizona Images — KNTV Production — Courtesy of Granite Broadcasting, Inc.,
Editor/Producer Bryan Foote.
Canyonlands — Courtesy of the National Park Service/Department of the Interior.
WCAU Fire Story — Courtesy of NBC-10, Philadelphia, PA.
Women in Sports – Paragliding — Courtesy of Legendary Entertainment, Inc.

Avid® iNEWS™ Newsroom Computer System Operations and Troubleshooting Manual

Part # 0130-06340-01 Rev B • 25 October 2004

# Contents


**Using This Guide**

Who Should Use This Manual . . . . . 1
About This Manual . . . . . 2
Symbols and Conventions . . . . . 3
If You Need Help . . . . . 4
...With the Syntax of Console Commands . . . . . 5
Related Information . . . . . 6
If You Have Documentation Comments . . . . . 6
How to Order Documentation . . . . . 7
Avid Educational Services . . . . . 7

Chapter 1 **Introduction**

What is Avid Unity for NEWS? . . . . . 10
iNEWS NRCS . . . . . 11
Third-Party Connectivity and Open Integration . . . . . 12
System Administrator Tasks . . . . . 13
Basic Tasks . . . . . 13
User Tasks . . . . . 14
Database Tasks . . . . . 14
Security Tasks . . . . . 15
Customizing the System . . . . . 15
Storage Maintenance Tasks . . . . . 15
Device Tasks . . . . . 16
Reviewing Default Settings . . . . . 17
Troubleshooting . . . . . 17

Chapter 2 **Connect Services**

Network Services . . . . . 20
Dialogs for Connect Services . . . . . 20
Building a Dialog . . . . . 20

Dialog Commands . . . . . 22
Dialog Examples . . . . . 22
Adding System Services . . . . . 23
Setting up the Service . . . . . 23
Console Connect Sessions . . . . . 27
Serial Connect Services . . . . . 28

Chapter 3 **Database Security**

Establishing Security Procedures . . . . . 30
User Passwords . . . . . 31
Checking Password Status . . . . . 31
Forcing Individual Users to Change Their Passwords . . . . . 34
. . . At an iNEWS Workstation . . . . . 34
Tracking User Activity . . . . . 35
Tracking User Login Activity and Date Created . . . . . 36
. . . At an iNEWS Workstation . . . . . 36
. . . At the Console . . . . . 36
Listing Users Currently Logged in. . . . . 37
Recording Logins . . . . . 38
Using Group Security to Control System Access . . . . . 39

Chapter 4 **Database Management**

Monitoring Free Space . . . . . 42
Understanding Database Storage Units . . . . . 43
Monitoring the Free List . . . . . 43
Understanding How the System Copies Stories. . . . . 44
Tracking Database Space over Time . . . . . 44
Using the hogs Command to Obtain Information . . . . . 45
Using dbserver and dbfree to Obtain Information. . . . . 46
Increasing Database Space for Immediate Use . . . . . 46
Maintaining the Database . . . . . 48
Checking the Database for Errors . . . . . 48
Cleaning the Database . . . . . 50
Cleaning Your Database Offline. . . . . 50

Chapter 5 **Backing Up the iNEWS System**


Tape Operations . . . . . 56
Establishing Policies for Backup Procedures . . . . . 56
Backing up the iNEWS Database . . . . . 57
The dbdump Command . . . . . 57
Backing up the Entire Database . . . . . 58
Backing up Individual Queues . . . . . 60
Notes on Backing up the Database . . . . . 62
Restoring Data to the iNEWS Database . . . . . 63
The dbrestore Command . . . . . 63
Restoring a First-Level Directory . . . . . 64
Listing Tape Contents and Backup Dates . . . . . 65
Listing Items Dumped on a Particular Date . . . . . 66
Listing the Date of Each Backup . . . . . 67
Searching a Tape . . . . . 68
The searchtape Command . . . . . 68
Searching a Tape for Stories . . . . . 69
Searching a Tape by Word(s) . . . . . 69
Searching a Tape by Word and Date Range . . . . . 70
Searching a Tape by Word and Day . . . . . 70
Searching a Tape by Word and Month . . . . . 71
Specifying a Maximum Number of Stories to Search . . . . . 71
Checking for Free Space on a Database . . . . . 72
Adding Blocks to the Free List . . . . . 72
Restoring the Database . . . . . 73
Disaster Recovery Planning . . . . . 73
Disaster Recovery Dbdump . . . . . 74
Create Minimal dbdump . . . . . 74
Backing up Software . . . . . 75
Backing up System Site Files . . . . . 76
The sitedump Command . . . . . 76


Chapter 6 **Disconnects**


Normal System Status . . . . . 80
Disconnects . . . . . 80

Detecting a Disconnect . . . . . 81
Types of Disconnect . . . . . 82
Causes of Disconnects . . . . . 83
Disconnect Recovery . . . . . 84
Overview . . . . . 84
Procedures . . . . . 85
Recovery Worksheet . . . . . 93

Chapter 7 **Troubleshooting**

iNEWS Workstation Problems . . . . . 96
A User Cannot Log in . . . . . 96
A User Cannot Establish a Session . . . . . 96
A User Cannot Access an Item . . . . . 98
Group Permissions . . . . . 99
Busy Stories . . . . . 99
Wire Problems . . . . . 101
Data Receiver . . . . . 101
PCU Wire Ports . . . . . 101
System Printer Problems . . . . . 103
Locked Blocks . . . . . 104
How to Check Process Status (ps Command) . . . . . 105
Power Failure . . . . . 106
Network Failure . . . . . 107
netstat -i Command . . . . . 108
Output Errors (Oerrs) . . . . . 108
Input Errors (Ierrs) . . . . . 109
iNEWS and FTS Connectivity (ftsdebug Command) . . . . . 109
Checking Communication between iNEWS and FTS Servers . . . . 110

Appendix A **Command References**

Programs Invoked by iNEWS . . . . . 114
Commands Used by Avid Personnel Only . . . . . 114
Linux Commands Used in iNEWS . . . . . 115
Console Control Commands . . . . . 116
Console Server Commands . . . . . 120

connect Command Options . . . . . 121
dbrestore Command Key Options . . . . . 129
dbtraits Command Options . . . . . 131
dbtraits Command Modes . . . . . 132
dbvisit Command Options . . . . . 133
dial Command Options . . . . . 134
grpcheck Command Options . . . . . 137
list c Command Options . . . . . 142
list d Command Options . . . . . 142
list q Command Options . . . . . 143
makeccutab Command Options . . . . . 147
makeshift Command Options . . . . . 148
maketab Command Options . . . . . 149
msgclean Command Options . . . . . 149
otod Command Options. . . . . 150
rename Command Options . . . . . 153
sitedump Command Options . . . . . 156
siterestore Command Options . . . . . 156
softrestore Command Options. . . . . 157
utraits Command Options . . . . . 160
utraits Command Flags . . . . . 160
Job List Commands . . . . . 163
Dialog Commands . . . . . 169

Appendix B **System Files**

/etc/dhcpd.conf . . . . . 178
/etc/hosts . . . . . 178
/site/config . . . . . 179
/site/printers/hplaser1 . . . . . 183
/site/system. . . . . 183
/site/wires/anpa7. . . . . 184
console.cfg . . . . . 185
SYSTEM.CLIENT.WINDOWS . . . . . 186
SYSTEM.CONFIGURE.301-ACTION . . . . . 187
SYSTEM.MAP . . . . . 188

SYSTEM.RESOURCE . . . . . 190
SYSTEM.WIRES.DISTRIBUTION . . . . . 192
SYSTEM.WIRES.KEYWORDS. . . . . 193
SYSTEM.WIRES.KEYWORDS-AP. . . . . 193
SYSTEM.WIRES.KEYWORDS-AP2. . . . . 194

Appendix C **Standard Dictionaries**

Using Dictionaries to Define Messages and Commands . . . . . 196
Names and Locations of Dictionary Files. . . . . 196
Customizing Dictionaries. . . . . 198
Changing Default Dictionary Values. . . . . 198
Restoring Dictionary Defaults . . . . . 201
Utility Messages Dictionary (/site/dict/messages) . . . . . 203
DBServer Program Messages . . . . . 204
Disconnect Program Messages . . . . . 205
Category and Keyword Check Program Messages . . . . . 205
Keyboard Check Program Messages . . . . . 206
Keyboard Check Program Messages for Macros . . . . . 207
Grpcheck Messages . . . . . 209
Wire Program Messages . . . . . 210
Mail Server Messages . . . . . 211
Validation (Action) Server . . . . . 211
Seek Server Messages . . . . . 211
Last Login Messages . . . . . 212
Messages for Print Server . . . . . 213
Message for the Sony Barcode Printer . . . . . 213
dbtraits Messages . . . . . 213
ccumsgs Dictionary (/site/dict/ccumsgs) . . . . . 213
PCU Messages . . . . . 214
Commands Dictionary (/site/dict/ccucmds) . . . . . 217
Commands(/site/dict/ccucmds) . . . . . 217
Job List Commands . . . . . 222
Queues Dictionary (/site/dict/queues) . . . . . 223
Queues Dictionary . . . . . 224
Words Dictionary (/site/dict/words) . . . . . 226

Connect Dictionary (/site/dict/doac) . . . . . 230
Telex Dictionary (/site/dict/telex). . . . . 234
Dial Dictionary (/site/dict/dial) . . . . . 236
Keyboard Macros Dictionary (/site/dict/keymacros) . . . . . 237
Printer Messages Dictionary (/site/dict/printmsgs) . . . . . 239
Case-shifting Dictionary (/site/dict/shift) . . . . . 240
MCS Dictionary (/site/dict/mcs) . . . . . 242
Device Types Used by Monitor Servers and Drivers. . . . . 242
Special Strings Recognized by the Monitor Server . . . . . 243
Error Messages for the Monitor Server . . . . . 243
Error Messages Shared by Monitor Servers and Drivers . . . . 246
Driver Errors Sent to MCT Only. . . . . 246
Driver Errors Sent to MCT &Possibly to Console & Mon User 246
Driver Errors Sent to User Only. . . . . 247
Status Reported in Device Status Field. . . . . 247

Appendix D **PCU Reference**

Overview. . . . . 250
PCUs . . . . . 250
Resetting a PCU . . . . . 251
PCU LED Displays. . . . . 251
PCU LED Codes . . . . . 252
Connecting Devices to a PCU . . . . . 253
Locating Ports . . . . . 253
Connecting a PCU to the System . . . . . 253

Appendix E **Character Maps**

ASCII (7-bit) Character Set. . . . . 255
IBM Character Set . . . . . 261
dbrestore Character Map (Arabic) . . . . . 268
Sample Arabic Wire Profile. . . . . 269

Appendix F **Environment Variables**

Registry Editor . . . . . 274
Environment Variables (Registry Values). . . . . 275

CCColor . . . . . 276
DestinationOrder . . . . . 278
MailLookup . . . . . 279
MsgMailAlert . . . . . 280
PIColor . . . . . 282
RGB Hexadecimal Color Chart . . . . . 283
ShowTimingBar . . . . . 284
Scan Codes . . . . . 285
SyncToServer . . . . . 288
Delete_Notify . . . . . 290
Environmental Variables for Servers . . . . . 291

Appendix G **Managing Traits at the Console**

Viewing User Traits from the Console . . . . . 296
Modifying User Traits from the Console . . . . . 297
Changing a Users' Password . . . . . 298
Listing Users Who Do Not Have Passwords . . . . . 300
User Traits Console Command Summary . . . . . 304
User Traits Summary . . . . . 305
Managing Database Traits from the Console . . . . . 309
Getting Basic Information . . . . . 309
Getting Detailed Information . . . . . 310
Changing Database Traits from the Console . . . . . 311
Changing a Parent Directory Only . . . . . 311
Database Traits Console Command Summary . . . . . 312
Database Traits Summary . . . . . 312
Sortfield . . . . . 321
Changing a Queue's Sort Field . . . . . 322
Starting the Queue Sort Function from the Console . . . . . 322
Purge Interval . . . . . 323
Abstract Printing . . . . . 324
Abstract Printer . . . . . 325
Abstract Lines . . . . . 325
Abstract Style . . . . . 326
Mailbox . . . . . 326

The dis Column . . . . . 327
Preview Lines . . . . . 327
Refresh . . . . . 328
Managing Group Traits at the Console. . . . . 329
Groups . . . . . 329
Read Group. . . . . 330
Write Group . . . . . 330
Notify Group . . . . . 330
Restricting Access Using Read and Write Limitations . . . . . 331
Removing Directory or Queue Restrictions . . . . . 332

Appendix H **The Line Editor, ed**

Launching ed . . . . . 334
Specifying Lines to Edit . . . . . 334
Searching the File . . . . . 336
Searching Tips. . . . . 336
Editing Commands. . . . . 337
Saving Changes. . . . . 340
Quitting ed . . . . . 341

**Index**

# Using This Guide

Congratulations on your purchase of an Avid iNEWS newsroom computer system. It is a vital part of the Avid Unity for News system solution that integrates with other Avid systems and third-party equipment to provide an ideal nonlinear workflow and optimize the news production process.

This manual provides information on how to manage the iNEWS newsroom computer system, consisting of:

- A console
- One or more servers
- Various clients, such as iNEWS Workstations and printers

*The documentation describes software features and hardware related to the iNEWS newsroom computer system, which is extremely customizable. Your system may not contain certain features and/or parts that are mentioned in the documentation.*

## Who Should Use This Manual

This manual is written for system administrators who are managing the iNEWS newsroom computer system (NRCS). It is strongly recommended that system administrators have prior experience in or classroom knowledge of Red Hat Linux system administration. All iNEWS system administrators need to:

- Manage user accounts, security and permissions
- Start up and shut down iNEWS NRCS
- Perform file system maintenance, backup, and recovery

- Maintain disks
- Monitor processes
- Configure and monitor the network

# About This Manual

This manual is part of a two-book set designed to keep pace with current advances in the Avid system's news production capabilities. The set—made up of this book and the *Avid iNEWS Setup and Configuration Manual*—is a comprehensive resource of all the information you will need to take advantage of the many options available to you. This manual will lead you through even the most complex procedures with task-oriented instructions. The information provided here builds on basic news production procedures described in the help system and other user-based guides, while adding a complete explanation of all the tools and techniques required to manage the newsroom computer system, including useful tips, shortcuts, and custom options.

The Contents lists all topics included in this guide. They are presented with the following overall structure:

- The Introduction helps you get oriented with beginning concepts and general workflow and provides valuable pointers to keep in mind as you proceed.
- The main body of this guide follows the natural flow of your work, with clear and comprehensive step-by-step procedures.
- Finally, a detailed Index helps you quickly locate specific topics.

# Symbols and Conventions

Avid documentation uses the following symbols and conventions:

| Symbol or Convention | Meaning or Action |
|---|---|
| | A note provides important related information, reminders, recommendations, and strong suggestions. |
| | A caution means that a specific action you take could cause harm to your computer or cause you to lose data. |
| | A warning describes an action that could cause you physical harm. Follow the guidelines in this document or on the unit itself when handling electrical equipment. |
| > | This symbol indicates menu commands (and subcommands) in the order you select them. For example, File > Import means to open the File menu and then select the Import command. |
| ▶ | This symbol indicates a single-step procedure. Multiple arrows in a list indicate that you perform one of the actions listed. |
| Margin tips | In the margin, you will find tips that help you perform tasks more easily and efficiently. |
| *Italic font* | Italic font is used to emphasize certain words and to indicate variables. |
| **`Courier Bold font`** | Courier Bold font identifies text that you type, such as commands entered at the console multiplexor. Output displayed on the console screen will appear in regular `Courier font`. |

| Symbol or Convention | Meaning or Action |
|---|---|
| | *Lengthy console displays may be edited to emphasize only the most important information in this manual. An ellipsis (...) represents portions of the console display not shown in the text. The console can display each server's prompt based on the system ID (typically a station's call letters) and the server's name. Examples in this manual use NRCS as the fictional station and system ID. For instance, the following is the console prompt for server A on the NRCS system:* `NRCS-A$` |
| Click | Quickly click and release the left mouse button. |
| Double-click | Click the left mouse button twice rapidly. |
| Right-click | Quickly click and release the right mouse button. |
| Drag | Click and hold the left mouse button while you move the mouse. |
| Ctrl+*key* | Press and hold the first key, such as Ctrl, Alt, or Shift, while you press the second key. |

# If You Need Help

If you are having trouble using iNEWS:

1. Retry the action, carefully following the instructions given for that task in this guide. It is especially important to check each step of your workflow.
2. Check for the latest information that might have become available *after* the documentation was published in one of two locations:
   - If release notes are available, they ship with your application CD, are in PDF format, and may be viewed from the CD using Adobe® Reader, a program freely available on the Internet.
   - If ReadMe files are available, they are supplied in your Avid application folder. ReadMe files are also available from Help.

*Release notes and ReadMe files are also available on the Avid Knowledge Center.*

3. Check the documentation that came with your Avid application or your hardware for maintenance or hardware-related issues.
4. Visit the Knowledge Center at www.avid.com/support. Online support services are available 24 hours per day, 7 days per week. Search this online Knowledge Center to find answers, to view error messages, to access troubleshooting tips, to download updates, and to read/join online message-board discussions.
5. For Technical Support, please call 800-800-AVID (800-800-2843).

   For Broadcast On-Air Sites and Call Letter Stations, call 800-NEWSDNG (800-639-7364).

## ...With the Syntax of Console Commands

If you are at the console, and are unsure about the function of a console command, use the `help` command.

To view instructions about using a command, type `help` followed by the name of the command. For instance, type **`help dbvisit`** for an explanation of the dbvisit command. The following data appears:

```
dbvisit -<d or v or i> -{r or m name] -[s] [block# ...]
'r' for read only
's' for "slow" to eliminate cache usage
'm' for machine name to disconnect
'i' to just validate isam files
```

*Because of the margin limitations of this manual, console command lines may appear wrapped to multiple lines. This does not necessarily indicate the need to press an Enter key. Unless otherwise indicated, console commands should be typed on a single line, allowing the computer to wrap the text whenever the command line stretches beyond the screen margin.*

See “Console Control Commands” on page 116 for more information.
Also, see the *iNEWS Newsroom Computer System Setup and Configuration Manual* “Types of Console Commands” on page 28.

# Related Information

The following documents provide more information about the iNEWS newsroom computer system:

- iNEWS NRCS Prep, Installation, and Upgrade Manual
- iNEWS NRCS Training Guide
- iNEWS NRCS Release Notes

The following documents provide more information about some of the other Avid products that integrate with the iNEWS newsroom computer system:

- iNEWS ControlAir Installation and Operations Manual
- iNEWS ControlAir User Manual
- iNEWS ControlAir Release Notes
- Avid Data Receiver Installlation and Operations Manual
- MOS Gateway Installation and Operations Manual
- MOS Gateway Release Notes
- Media Browse Installation and Operations Manual
- Media Browse Training Manual
- Media Browse Release Notes

*For the latest information on all Avid products and documentation on other Unity for News products, such as the Avid AirSPACE video server, Avid Active ContentManager, or the Avid NewsCutter nonlinear editing system, see the Avid Knowledge Center:* `http://support01.avid.com`.

# If You Have Documentation Comments

We'd appreciate any comments or suggestions you may have about this document or any other piece of documentation. Please restrict your comments to documentation issues.

Please e-mail your documentation comments to:

TechPubs-WI@avid.com

Include the title of the document, its part number, and the specific section you are commenting on in all correspondence.

## How to Order Documentation

To order additional copies of this documentation from within the United States, call Avid Sales at 800-949-AVID (800-949-2843). If you are placing an order from outside the United States, contact your local Avid representative.

## Avid Educational Services

For information on courses/schedules, training centers, certifications, courseware, and books, please visit www.avid.com/training or call Avid Sales at 800-949-AVID (800-949-2843).

# Chapter 1

# Introduction

As part of the Avid Unity for News broadcast solution, the iNEWS newsroom computer system is an integrated, digital news creation, and production system. It provides journalists, producers, directors, writers, and technical personnel in a newsroom with an array of tools to make their jobs easier.

This chapter contains the following main sections:

- What is Avid Unity for NEWS?
- System Administrator Tasks

# What is Avid Unity for NEWS?

The Avid Unity for News system is a nonlinear shared storage media network that enables the immediate and simultaneous availability of footage upon acquisition to both journalists and editors; the sharing of footage among teams for shot selection, script writing, and production; the parallel editing of news packages and promotions; and on-air playout, without dubbing back to tape, within moments of story completion. It is a truly integrated broadcasting solution that minimizes lag between processes, where even activities such as asset management and Web updating become concurrent tasks that facilitate production, without interfering with the primary objective of putting the news on the air.

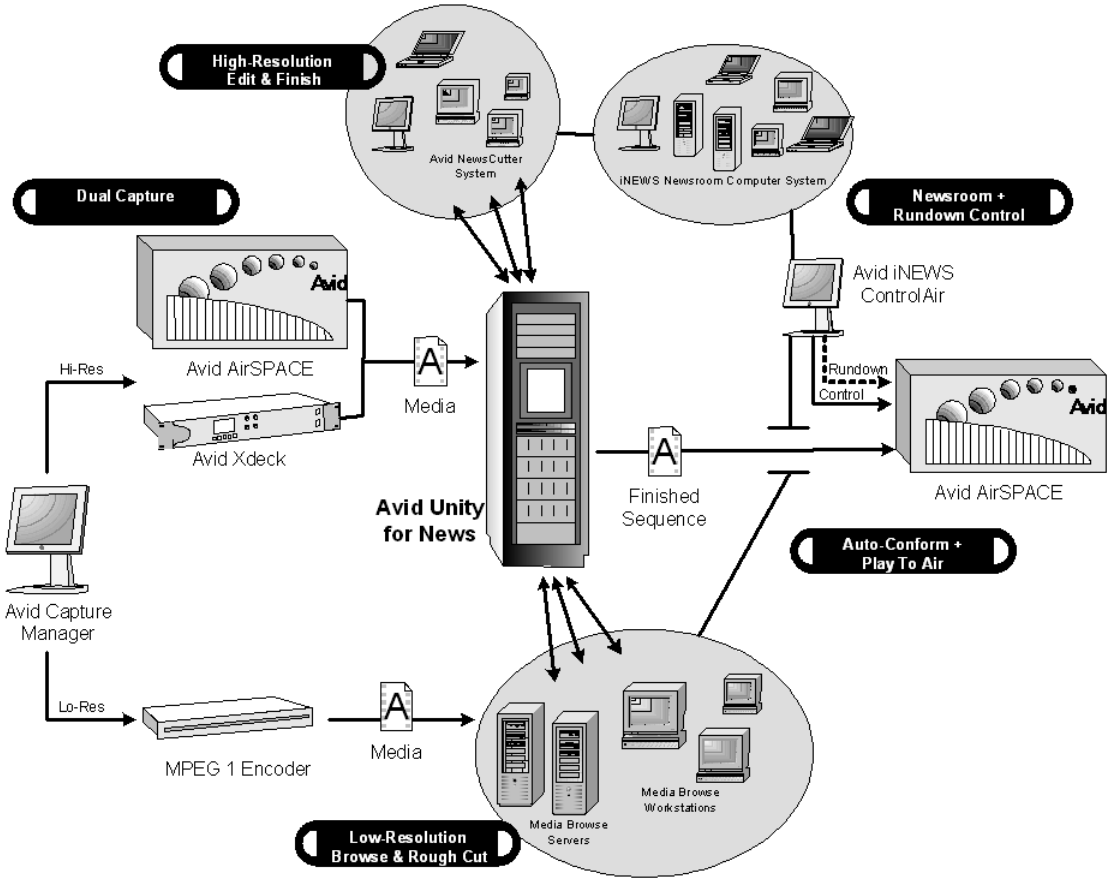
In the Avid Unity for News workflow, the Avid iNEWS newsroom computer system provides up-to-date news data throughout the newsroom and seamlessly links all tasks in the production process, such as wire service, assignment desk, script writing, browse, editing, rundown, playout, and device control tasks.

The Avid Unity for News environment consists of integration between several Avid products, such as:

- The Avid iNEWS newsroom computer system
- The Avid Media Browse system, including Avid Capture Manager
- The Avid iNEWS ControlAir system
- The Avid NewsCutter nonlinear editing system
- Avid Active ContentManager
- Avid AirSPACE video server(s)

Your newsroom may have any one or all of these products and others not mentioned in this brief summary.

The figure below shows a typical high- and low-resolution workflow, from capture to playout, in a shared storage environment with an iNEWS newsroom computer system:




## iNEWS NRCS

An iNEWS newsroom computer system (NRCS) provides:

- News gathering from video, audio, and text sources
- News production, including:
  - Story creation and script editing
  - Show planning and creation

- News to air, including:
  - On-air playback control
  - File exporting
  - Internet publishing

In an Avid Unity for News environment, Web production processes can occur in parallel with newcast production.

For more information, see “NRCS Components” in Chapter 1 of the *iNEWS Installation and Upgrade Manual.*

Some primary components of iNEWS NRCS include:

- Linux-based computers running the iNEWS Server software. In this manual, these host computers are referred to as the iNEWS Servers, or individually as server A, server B, and so forth.
- An IBM/Intel-compatible personal computer (PC) running the iNEWS console multiplexor program. This PC is known as the console. See Chapter 2 of the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information.
- Windows-based computer running the iNEWS client software. These PCs are known as iNEWS Workstations
- Other clients, such as printers.

## Third-Party Connectivity and Open Integration

Avid’s commitment to open integration and conformance with industry-standard technologies is clearly demonstrated as a major feature of the Unity for News product set. Avid supports open integration by, among other methods, offering no fewer than 13 Applications Programming Interfaces (APIs) for Avid products to interact with third-party devices, software, and custom development, such as the general NRCS-MOS protocol agreement, or Transfer Manager APIs.

The iNEWS system is capable of interfacing with a wide variety of major control systems and servers with dedicated device interfaces, including Omnibus™ and Sony®. The iNEWS system has an efficient Active-X implementation of the MOS protocol, which greatly extends its interfacing abilities. For instance, in conjunction with iNEWS, ControlAir provides a

central point of control for up to 32 production devices, such as video servers, character generators, and still store machines from third-party manufacturers; a sample of which is provided in Table 1.

**Table 1 Third-Party Integration**

| Device Type | Manufacturers |
|---|---|
| Video Servers | Sony, Grass Valley™, and Quantel |
| Character Generators | Chyron, Aston, Pinnacle, Abekas, Quantel, and Pixel Power |
| Still Store Machines | Leitch® and Quantel |

The Avid Unity Transfer Manager productivity tool synchronizes with the iNEWS rundown to determine file transfer priority—the story that appears sooner in the rundown will automatically be ranked higher in the queue, and playout order of the newscast matches the rundown precisely.

In addition to conforming to accepted, standard file formats, such as Open Media Framework (OMF), Avid systems accept media in more than 30 other file formats, allowing them to accept, use, track, and present any media, images, graphics, audio, scripts, or notes.

# System Administrator Tasks

The following sections introduce many system administrator responsibilities and the system's capabilities and functions.

## Basic Tasks

Before you can customize or maintain the iNEWS newsroom computer system, you must learn several basic tasks, which include:

- Start up or shut down iNEWS NRCS, which includes logging out users and taking the system offline.

- Backing up a site file before making file modifications.
- Send system administrator commands from the console to one or more of your system's computers.
- Be a console superuser, capable of setting up special superuser permissions.

## User Tasks

A user is anyone who can log in to the database and use iNEWS NRCS. Your responsibilities regarding users are:

- Monitor user information, such as users' access privileges and which users are currently logged in.
- Customize the traits of users' accounts to enable users to more effectively use the system.
- Provide a new employee access to the information stored in the iNEWS NRCS database by creating a new user account.
- Remove user accounts of former employees to prevent improper access to the iNEWS NRCS database.

## Database Tasks

The iNEWS system database contains the information your oganization needs to function. A system administrator's tasks associated with the database include:

- Design forms (that is, story templates) to display important information about stories in a queue.
- Monitor changes to files and queues in the database.
- Unlock or delete any item in the database, and recover items that were accidentally deleted or corrupted.
- Create new folders or queues in the iNEWS system database to meet your organization's expanding needs.
- Remove a directory or queue from the database, if it is no longer used.
- Change the name or traits of an existing directory or queue.

- Assign the mailbox trait to queues for configuring automatic story distribution into and out of queues.

## Security Tasks

There are many ways to ensure the security of your iNEWS system. Your responsibilities regarding system security include:

- Monitor and change passwords or force users to change them by setting up system checks and modifications.
- Monitor user login activity to guard against unauthorized use of the iNEWS system.
- Assign security to a directory or queue, limiting access to a specific group of users.
- Restrict database access by placing users into security groups based on job roles and need for information.

## Customizing the System

Your responsibilities regarding customization include:

- Customize command names, message text, and other items by changing their entries in your system's dictionary files.
- Remove your custom dictionary translations by reverting to the default settings for command names, message text, and other items.
- Create templates for the CG Title Entry tool.
- Design and assign custom keyboards for users with a unique set of keyboard macros.

## Storage Maintenance Tasks

You will want to monitor the database regularly to ensure adequate storage. Storage maintenance tasks include:

- Monitor how much free space is available in the database and, if necessary, increase the amount to prevent the system from running out of space.
- Perform preventive database maintenance by periodically running certain utility programs that can find and fix minor problems before they become serious.
- Backup the entire database or portions of it onto tape, so if necessary, the information can be restored to the database later.
- Make a backup copy of files, such as the site file, on tape any time you make important changes.

## Device Tasks

A device is any kind of hardware or software that performs a specific function when it is set up on the iNEWS system. Your responsibilities regarding devices include:

- List the parameters of any device running on your system or list all devices of one type.
- Add any type of device to your system, if you have the capacity and license permission.
- Edit site-specific files, to change the setup information for a device in your system's configuration file.
- Reconfigure the system so it recognizes any changes you make to your system's devices.
- Setup printer styles so that different tasks can be printed in pre-designated formats.
- Setup servers which are utility programs that automatically perform various actions on the stories in your database.
- Change wire distribution and sorting of data coming into your database from a wire service to queues based on their category codes or content.
- Write dialogs (lists of instructions) for each service to automate the connection process. A service is a device that connects a user to a remote computer system.

## Reviewing Default Settings

Your responsibilities regarding system profiles, default settings, and command syntax include:

- Changing a system profile setting to change your system's operation.
- Reviewing default settings of all system profile parameters.
- Reviewing command syntax for edit, console, and job list commands.

## Troubleshooting

Your troubleshooting responsibilities include:

- Transfer system activities from a halted computer to other system computers. If a computer connected to the system has been halted, bring the system back to operation using the remaining computers.
- Reconnect a computer that has been halted. Following routine maintenance, reintegrate a computer into your system's operation.

# Chapter 2

# Connect Services

Connect services are utilities you can set up to enable iNEWS users to connect to other computers or information services over the network, such as a remote console for your iNEWS newsroom computer system.

This chapter contains the following main sections:

- Network Services
- Adding System Services
- Console Connect Sessions
- Serial Connect Services

# Network Services

It is useful to set up a service to allow you to connect to your iNEWS Servers. For instance, setting up network remote connect services allows you to perform many administrative tasks from your iNEWS Workstation rather than at the console.

## Dialogs for Connect Services

A *dialog* is a script that tells the service what prompts to expect from the device it connects to, and provides the service with appropriate responses for each prompt. Using any service involves some routine activities, such as logging in, that you do each time you use that service. You can create a dialog for any service to handle these routine activities.

## Building a Dialog

This section explains how to design and create a simple dialog that logs a user in to an information service. Once logged in, the dialog yields control to the user until he/she is ready to close the connection. Then the dialog logs the user out of the service and closes the connection.

When you design a dialog, it is helpful to turn on the `capture connect` command and perform the procedure you want to incorporate into the dialog. This way, steps in that procedure are captured to a story that you can refer to while designing the dialog; each line in the dialog is built as a line of the story.

This process includes the most commonly used dialog commands.

**To build a dialog:**

1. Create a dialog queue.

   Each dialog must be in a separate queue in the `SYSTEM.DIALOG` directory. The first step in creating a new dialog is to go to `SYSTEM.DIALOG` and create a queue to hold it. The name you give this queue is also the name of the dialog, so choose a queue name that describes what the dialog does, such as Console, for a dialog used

during connect sessions to the console. See the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information about adding new directories, queues or stories.

2. Open a new story in the queue and build the dialog.
3. Type a message that will be displayed on the user's workstation, indicating the connection is being established, using the following format:

   **`message <"text">`**

   The `message` dialog command displays on the user's screen whatever text you put in the `<"text">` parameter.

4. Type:

   **`wait CONNECT`**

   When you use a modem to make the connection, you want the dialog to wait until the modem sends the word `CONNECT` to the workstation, signaling it has made the connection.

   If necessary, use these additional commands to accommodate the login procedure:

   | | |
   |---|---|
   | `delay nn` | Use this command to pause the dialog while the service continues connect and login processes.<br><br>(`nn` is the number of seconds you want the dialog to pause before continuing to the next command) |
   | `type <"text">` | Use this to send a message to the device the workstation is being connected to.<br><br>`<"text">` includes the message |

5. To send your login data, such as an account number, type:

   `type <"text">`

6. Type **`pass`** to yield control to the user after login is complete.

   This command instructs the dialog to pass whatever the user enters to the device to which the service has connected.

Use `pass x` to include a character the user can enter to cue the dialog to resume. The *x* should be a character or symbol, such as the at symbol (@), he or she will not normally use.

Use `pass` alone to instruct the system to accept what the user enters until he/she closes the connection.

7. Type commands to log out of the service.

   When the user exits a connection using the `quit connect` command, the dialog resumes and performs the logout process and closes the connection.

8. Attach the dialog to the service by placing the dialog's name in the service's `dialog` parameter in the service table, located in the `SYSTEM.SERVICE` queue.

9. Type the **`configure -s`** command to incorporate this change into the service.

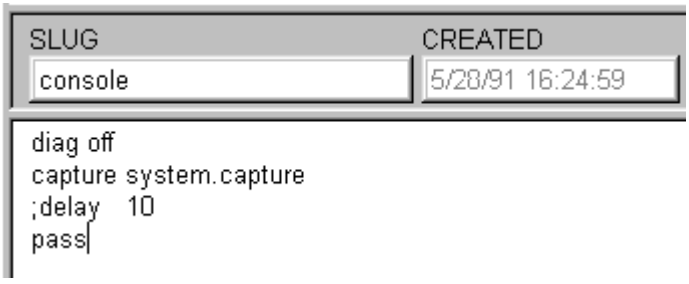
## Dialog Commands

Available dialog commands are flexible enough to script entire connect sessions. For instance, a dialog could be constructed to automatically log in to the remote console and "unbusy" a rundown if desired.

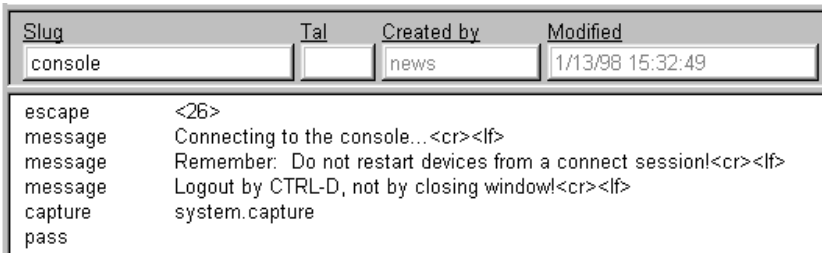
The available dialog commands are reviewed in Appendix A.

## Dialog Examples

Here are a few examples of dialogs for remote console connect services:



In the example, <cr> and <lf> are used to indicate a “carriage return” and “line feed.”




# Adding System Services

A service consists of two parts:

- A network resource, set up as a device in the configuration file (`/site/config`).
- The service, set up in the service table in the database file (`SYSTEM.SERVICE`); it uses the resource to make a connection.

  To add a new service to your system, you must design and build the service and select a network resource for the service to use. If an appropriate resource does not exist, create one.

## Setting up the Service

**To set up the service and its resource:**

1. See if the service and its resource exists.

   Try connecting to a service you want. If it works, then the necessary programs are installed. If you cannot find the service, or one does not work, call Avid Customer Support for help.

2. Choose a device number for the resource and a name for the service. In the following examples, the service name is `console`.

   Check your configuration file (`/site/config`) to determine the appropriate device number for your new resource. For a network resource, choose a number in the range you have reserved for services, servers, and similar devices. Ensure the number you choose is out of the range used for normal devices.

See the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information on selecting servers and the Site Configuration file.

3. Add the resource to the configuration file on each server in your system.

   Add the device number of the network resource to the `reslist` line in the server's host definition. To keep the network resource available if you bring down one of the system servers, add this line to any appropriate alternate `host` definitions.

   a. Select *all* servers.

The following steps use `ed` to modify system files. If you are unfamiliar with this line editor, see Appendix H.

   b. Use `ed` (line editor) to open the configuration file. The display will look similar to:

```
NRCS-A$ ed /site/config
1356
```

   The general format for a network resource configuration line is:

```
resource <device #> <resource name> <device name>
```

| Parameter | Description |
|---|---|
| `device #` | Identifies resource's device number. |
| `resource name` | Name you want to give to the resource. Network resources can share the same name; services choose the first available resource of the correct kind. In the following example, we call the resource `net`. |
| `device name` | Resource's device name. If you do not want to give it a name, put a hyphen in this position. |

   For instance, our new network resource configuration line would look like this:

```
resource   220  net - ; net connect
```

4. Add the service to the service table in the database.

   Each service installed in your system is defined on a separate line in your system's service table, which is the first story in `SYSTEM.SERVICE`. Each service defined in the service table consists of a few parameters that determine how the service behaves. It does not matter where in the service table you add the new line.

A service line has six parameters in this general format:

```
<service> <host> <dialog> <resource> <group> <command>
```

The following table contains information on service line formats:

| Parameter | Definition |
|---|---|
| `service` | Name you want the service to have. |
| `host` | Name of the server where you want the service to look for its resource. List the target server's name in your system's `/etc/hosts` file.<br><br>Normally, put a hyphen in this position of the service line to force the service to search each server until it finds one that has a resource it can use. |
| `dialog` | If you want the service to use a dialog, put the dialog name in this position of the service line. Otherwise, fill this position with a hyphen. |
| `resource` | Name of the resource that you want the service to use. |
| `group` | You can restrict who can use the service by specifying a user group. If you do not want to restrict access, place a hyphen in this position. |
| `command` | Specifies the command you want the service to use to handle communication. To create a service that establishes a network connection, type the `telnet` command here. Ensure you specify the program's full pathname and include the system name to which you are connecting. |

You will need one such service line for each server on your system. Use the `rlogin` command in the following way:

```
/usr/bin/rlogin <servername> -1 <username>
```

The following sample entries in `SYSTEM.SERVICE` let you establish a network connection to your console from an iNEWS Workstation session.

```
a - console console sysops /usr/bin/rlogin NRCS-A -l so
b - console console sysops /usr/bin/rlogin NRCS-B -l so
```

The resource assigned to the service must be configured on the computer you want to log in to.

5. (Optional) Create a dialog for the service.

   If you want your service to follow certain instructions every time it is invoked, create a dialog for the service that contains those instructions. See "Building a Dialog" on page 20 for more information.

6. (Optional) Test your configuration changes on iNEWS. See the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information on testing the site configuration file.

7. Reconfigure the system.

   This causes your system to note changes and incorporate them into appropriate programs. Do the following:

   See the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information on selecting servers and the superuser mode.

   a. Select the master computer (typically server A).

   b. Enter superuser mode.

   c. Type:
      ```
      NRCS-A# offline
      ```

   d. If you added a new resource or modified an existing one in the process of creating a new service, reconfigure your system by typing the following:
      ```
      NRCS-A# configure
      ```

      If you modified an existing service or added a service that uses an existing network resource, you did not make any changes to your system's configuration file. So, you need to have the system note only changes made to the service table by typing the following:
      ```
      NRCS-A# configure -s
      ```

8. When the prompt returns, type:
   ```
   NRCS-A# online
   ```

9. Exit superuser mode by holding the Control key (Ctrl) down and typing the letter D. A message similar to the following will appear:
   ```
   A Tue Oct 5 00:18:58 2000 msg System being configured
   ```

   When you see the system prompt, the network service you created and its resource you added (if any) is ready for use.

10. (Optional) Back up site files.

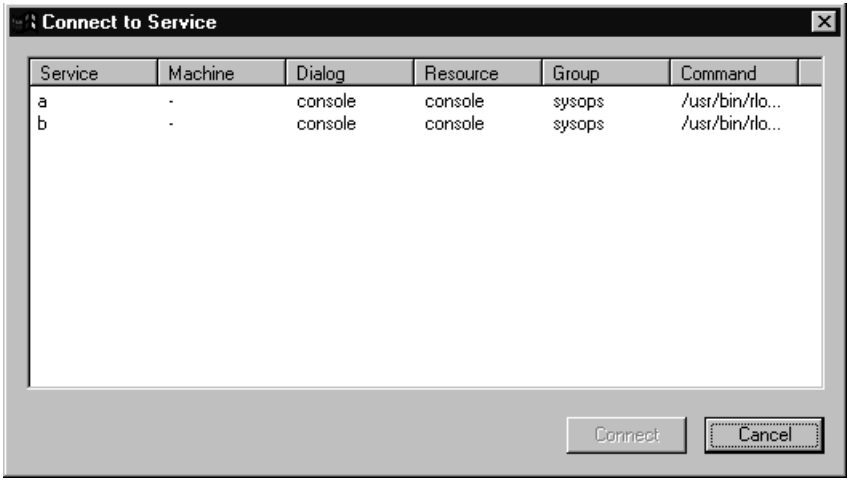
If you have made significant changes, back up your site files with the `sitedump` console command.

# Console Connect Sessions

To connect to the console from an iNEWS Workstation—that is, once the network remote connect service and its resources are set up—do the following:

1. Log in to any iNEWS Workstation, with a user account allowed to use the Connect to Service feature.
2. Select Communicate > Connect to Service. The Connect to Service dialog box will appear, offering you a list of services.

   As shown in this example, two options are provided to connect to the console—each option corresponding to a different iNEWS Server.

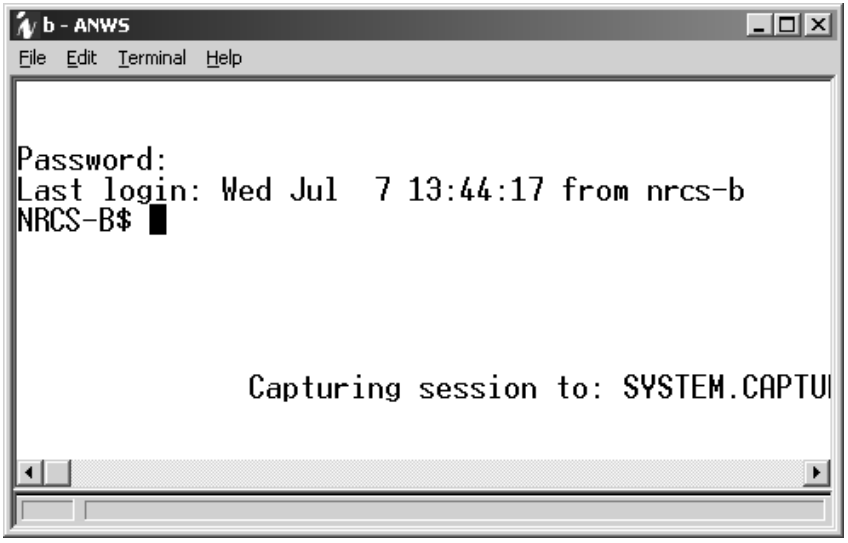



3. Double-click on the service you want.

   A dialog box will appear with a dialog (as defined in the `SYSTEM.DIALOG`) that requests a password.
4. Type in the password and press Enter.

5. Once connected, the dialog box will display the console prompt, similar to what appears on the console when you are logged in as a system operator.

   Here is an example of a remote console connection:




6. You can now perform various administrative tasks through the console connect session, as opposed to doing them while physically located at the console.

**To stop a console connect session, press Ctrl-D. Do not close the dialog box by selecting the Close option from the File drop-down menu or by clicking on the X button in the upper-right corner.**

# Serial Connect Services

Although less commonly used since the advent of the Internet, serial connect services can also be configured to connect and log in to other computers, either via modem dialup or direct serial line.

Contact Avid Customer Support if you need to configure a serial connection for your users.

# Chapter 3

# Database Security

This chapter describes how to use various features in the iNEWS newsroom computer system to establish and maintain database security.

This chapter contains the following main sections:

- Establishing Security Procedures
- User Passwords
- Checking Password Status
- Tracking User Activity
- Using Group Security to Control System Access

# Establishing Security Procedures

Use the following guidelines to improve the security of your system:

- Set up official security procedures and have everyone in the newsroom follow them.
- Keep track of your backup tapes. You can get user passwords from a backup tape.
- Assign users superuser status only when they need it.
- If any user does not need superuser status, remove it. Create two user accounts for staff members who need superuser privileges—one to be a superuser and another to be a regular user. That way, you can track activity of superuser accounts.
- Change the superuser password regularly.
- Use a security modem for some of your system's dial-up connections. A security modem provides an additional layer of password protection.

*Security modems may not work with laptops. Contact Avid Customer Support for more information about security modems.*

- Ensure that users do not use their names, station call letters, or other easily guessed words as passwords. Require everyone to include at least one non-alphabetic character such as a punctuation mark or a number.
- Use the system profile to set a required minimum length for all passwords so no one uses a short password.
- For devices for which you have dedicated resources—that is, devices that have a one-to-one correspondence between the physical device and device number used to identify it—use device name security on workstations in specific locations and put the device ID in a group.
- If you have dedicated resources, you can put a MODIFY-DEV field in the queue's story form if you suspect that someone has broken into a user's account. When changes are made to stories created after the MODIFY-DEV field has been added, the system puts the device name of the workstation where changes were made in that field. This information is kept internally, but having the MODIFY-DEV field makes it visible to users.

- Be familiar with valid user accounts on your system. Restrict access to these account names to protect against unauthorized outside access.

# User Passwords

Your iNEWS newsroom computer system protects against unauthorized access by giving each authorized user a password to log into the system. Group security, described under “Using Group Security to Control System Access” on page 39, lets you control specific areas of your database that each user can access.

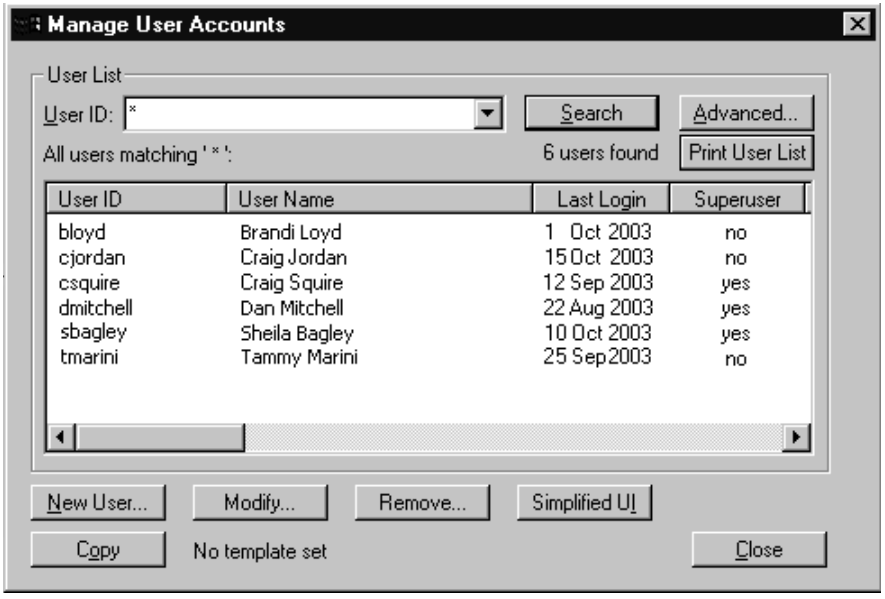
# Checking Password Status

A user account without a password is an open door to your system. You should always give a user a password when you add the user to your system. See the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information on setting up new user accounts.

For information on how to check password status from the console, see Appendix G.

However, if you suspect that a user does not have a password or has not changed it in awhile, you can find out for certain from any iNEWS Workstation. To do this, log in as a system administrator—that is, use a superuser account—and do the following:

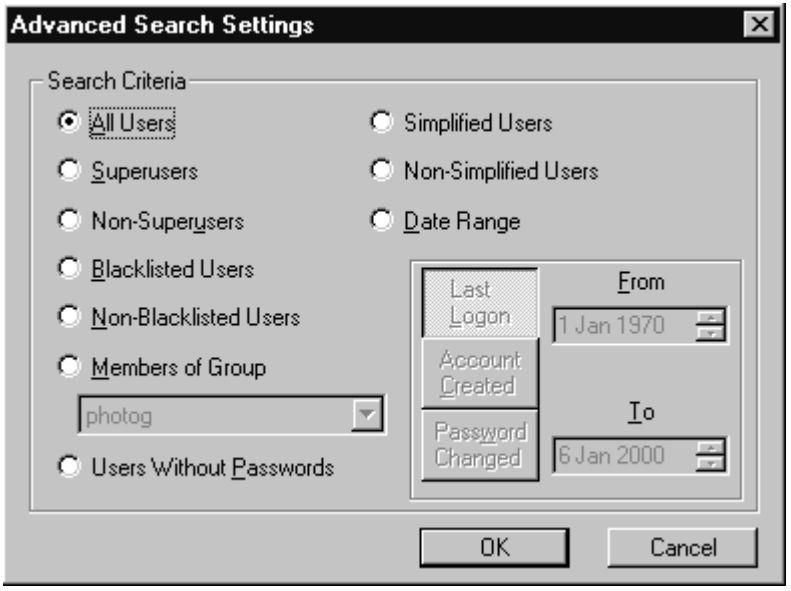
1. Select Tools > Options > Users.
2. The Manage User Accounts dialog box appears.




3. Type an asterisk (*) in the User ID field if it does not already appear. The iNEWS system recognizes this as a wildcard and therefore will search the entire database of users.

The other criteria options available in this dialog box are explained in Chapter 4 of the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

4. Click the Advanced button. The Advanced Search Settings dialog box appears with All Users selected by default.




5. Do one of the following:
   - To search for all users without passwords, select Users Without Passwords.
   - To search for all users who have not changed their password within a specific time, select Date Range. Then click Password Changed, and specify the time frame to search.
6. Click OK to confirm your advanced search setting or click Cancel to cancel it.
7. Click Search to initiate the search.

A progress bar appears if a lengthy search is underway. Results of the search appear in the User List field in the center of the Manage User Accounts dialog box.

As a system administrator, you can change a user's password. Change a user's password to provide a new user with a temporary password or to supply an established user with a new password if the user forgets it and cannot log in. See the *iNEWS Newsroom*

*Computer System Setup and Configuration Manual* for more information on how to do this from any iNEWS Workstation. For steps on how to change a user's password from the console using the `utraits` console command, see "Changing a Users' Password" on page 298.

## Forcing Individual Users to Change Their Passwords

Occasionally, you may have individual users who do not change their passwords as required. When that happens, you can force them to change their password at their next login. You can do this for a single user at an iNEWS Workstation.

To force multiple users to change their passwords—such as all users who haven't changed their passwords in the past six months—you must go to the console. For steps on how to force password changes from the console, see "Changing a Users' Password" on page 298.

### . . . At an iNEWS Workstation

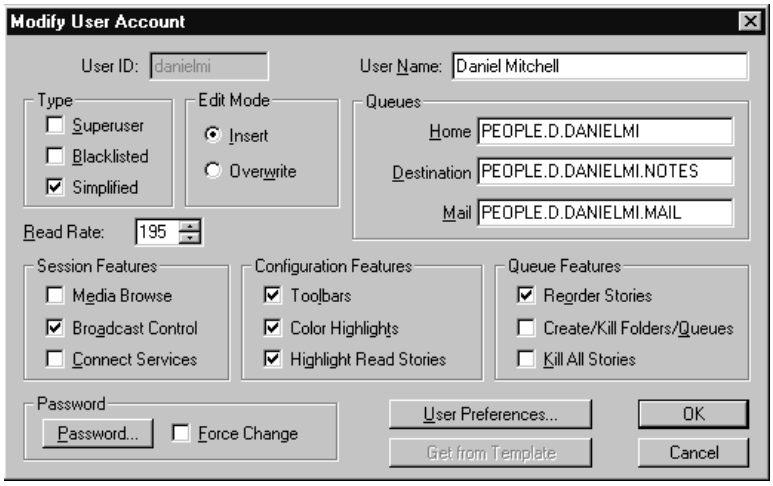
**To force an individual user to change his or her password:**

1. Select Tools > Options > Users.

   The Manage User Accounts dialog box appears.

2. Type the user name in the User ID field.
3. Click Search.
4. Select the user name when it appears in the dialog box.
5. Click Modify.

The Modify User Account dialog box appears.




6. Check the Force Change box.
7. Click OK.

   The next time the user logs in, he or she will be required to choose a new password.

# Tracking User Activity

Keep a record of who uses iNEWS and when they use it by following the procedures described in this section. These security measures can ensure that there is no unauthorized use of your system.

At an iNEWS Workstation, you can determine:

- Last login date of one or more user accounts
- Date user accounts were created
- Users currently logged in

At the console, you can determine attempted and successful logins.

# Tracking User Login Activity and Date Created

You can search for a user account's last login and the date the user account was created from an iNEWS Workstation.

## . . . At an iNEWS Workstation

**To search for a user's last login:**

1. Select Tools > Options > Users.

   The Manage User Accounts dialog box appears.

2. Click Advanced. The Advanced Search Settings dialog box appears.
3. Click Date Range.
4. Do one of the following:
   - To search for user accounts with a last login date that matches a specified date range, click Last Login.
   - To search for user accounts created within a specified date range, click Account Created.
5. Specify a date range in the From and To fields.
6. Click OK.
7. Click Search.

   The requested user name information appears.

## . . . At the Console

Another command you can type at the console will give you valuable information about users:

```
list u-t [<username>]
```

This command shows the date and time a user account was created, date and time of last login, and date and time of last password change.

If you do not specify a particular user name to check, you will get a listing for all users.

## Listing Users Currently Logged in

From the iNEWS Workstation, you can see a list of all users logged in, and you can find out whether a specific user is currently logged in.

**To see a list of all logged-in users:**

1. Select Communicate > Messages > Logged In Users.

*You need to be sending a new message to use this function. Begin a new message (to anyone) to obtain access to the menu items in this step.*

A dialog box appears with a list of users currently logged in.

2. Click OK when you are done looking at the list.

**To determine whether a specific user is logged in:**

1. Select the Message bar.

2. Type the user name in the To field.
3. Move cursor to the message field, using the mouse or Tab key.
4. One of three symbols appears to the left of the To field, depending on what you type in the To text box:

If the user is currently logged in, an icon of connected cables appears to the left of the user name.

If the user is not currently logged in, an icon of disconnected cables appears to the left of the user name.

If there is no such user name in the system, a question mark appears to the left of the user name.

# Recording Logins

Keep track of successful and attempted logins to preserve system security. You can spot unauthorized users, people logging in at odd hours, or repeated attempts to guess passwords.

**To monitor logins from different types of devices:**

1. Change the value for the `W_LOGTYPES` token in the dictionary `/site/dict/words`. A typical definition for this token might look like this:

   ```
   W_LOGTYPES   /G
   ```

   Each letter in the `W_LOGTYPES` value represents a different type of device that can log in on an iNEWS newsroom computer system.

2. To track logins by device type, add the appropriate letter to the `W_LOGTYPES` value.

   The letters are the same ones that appear in the first column of a `list s` or `list c` display for these device types.

   Use the following letters:

   | | |
   |---|---|
   | G | iNEWS Workstation |
   | B | Web sessions |
   | c | COM |
   | r | Remote search |

   For instance, to track logins from all devices on your system, change the `W_LOGTYPES` line to look like this:

   ```
   W_LOGTYPES       /GBcr
   ```

   The letters can appear in any order.

*On a busy system, this can create a large amount of console activity.*

3. When a user logs in at a device of a type listed in `W_LOGTYPES`, a message similar to the following is sent to the console:

   ```
   G602: 12:04:34 Login arlin 192.168.1.100 2.0.0.157
   ```

This message includes the device type and number, time of login, user name, IP address of PC used, and client version number.

4. A logout message similar to the following is sent to the console when the user logs out:

   ```
   G602: 12:21:52 Logout arlin 192.168.1.100 2.0.0.157
   ```

   A failed login—that is, an invalid user name or password—produces a message similar to this:

   ```
   G602: 12:04:34 attempted login arlin 192.168.1.100
   2.0.0.157
   ```

   Regardless of the `W_LOGTYPES` value, a message is *always* sent to the console when a superuser logs in or out. This message includes `(n)` if the user is a superuser. An unsuccessful login by a superuser generates a message only if the device type used is included in `W_LOGTYPES`.

   Using login tracking in conjunction with console history and disk logging, you can keep accurate records of who is using your system and when and where they are connecting to it.

   A listing of currently logged in users may be obtained from the console using the list s command. See “list s” on page 144 for more information.

# Using Group Security to Control System Access

Your system is designed to be used by a wide range of people. For the system to accommodate so many diverse job roles, restrict sensitive areas of your database to authorized users. The system provides a powerful security feature that lets you restrict access to important directories and queues.

**Security ensures that only authorized people can view or make changes to important queues. It does not provide absolute privacy, because superusers can open any story while performing normal system maintenance. Warn your staff not to store personal or confidential material in the database.**

Many security features in iNEWS involve establishing groups and assigning privileges and restrictions to them. Chapter 6 of the *iNEWS Newsroom Computer System Setup and Configuration Manual* gives you complete information about using the group features in iNEWS to help you maintain system security.

# Chapter 4

# Database Management

This chapter provides you with information required to manage the iNEWS database, where all *business data*, such as stories, is stored. Much of the iNEWS *system data* is stored in configuration files, which is covered in the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

This chapter contains the following main sections:

- Monitoring Free Space
- Tracking Database Space over Time
- Increasing Database Space for Immediate Use
- Maintaining the Database

# Monitoring Free Space

The iNEWS newsroom computer system is constantly collecting wire stories and adding them to the database while the news staff adds scripts, rundowns, memos, assignment sheets, and other stories. To keep from running out of disk space, iNEWS tracks old stories. As stories get old, the system purges them.

Distribution and purging of disk space is called the *database cycle*. In this cycle, wire stories are collected and stored in the database for a specific interval of time. Other news items, such as scripts and assignment sheets, are created by the news staff and also kept for a preset period of time, called the *purge interval*, which is set individually for each queue. Any story older than its queue's purge interval is purged, and its space is reclaimed for new stories.

The purge interval is a database trait, so you can set different purge intervals for queues and directories, depending on the information they hold. Setting purge intervals appropriate for stories in various queues helps keep the database from growing too large.

Once an hour, at 15 minutes after the hour, an automatic `dbpurge` program scans each queue for stories older than the queue's purge interval and moves these stories into the Dead queue.

Stories sent to the Dead queue are not erased until the system needs the space. Until the system reclaims this space, stories in the Dead queue can be read, searched, edited, copied, or printed. To retrieve a story from the Dead queue, select the story and copy it to a different queue in the database, where you have write permission.

Although you can open stories in the Dead queue, they are marked for removal and will be permanently removed when the system detects the computer is running out of storage space. Your system keeps track of the space available by examining and maintaining a list of free space on the disk. The free list is explained in "Monitoring the Free List" on page 43.

## Understanding Database Storage Units

Your computer's disk is divided into blocks. The database portion of the disk is divided into 1024-byte blocks. When a story is saved, the system allocates as many blocks as necessary to hold the story and then divides the story among those blocks.

Blocks used to hold a story need not be sequential; a story can be saved in blocks that are apart. To tie together all blocks, each block contains a reference to the block containing the next part of the story.

A block is either "in use" or "free".

## Monitoring the Free List

The free list is the list of free space on the disk. By keeping track of space in the free list, the system can detect when it is running low on space. It runs an automatic `dbserver` program, which removes the oldest stories from the Dead queue and adds the space to the free list, where it is made available to the system. This way, `dbserver` maintains the volume of free space available in the database.

The free list measures space in *blocks* (a block=1024 bytes), and has a lower limit called the *low watermark*, which represents the least amount of free space available. When the space available drops below the low watermark, the system runs `dbserver` to reclaim enough space from the Dead queue to rebuild the free list to the *high watermark*.

Together, the high and low watermarks determine the free list's normal operational range. If the system cannot get back up to the high watermark after reclaiming free space, the user will get a `low on space` message. Create free space immediately, as explained in the following section.

## Understanding How the System Copies Stories

You can configure iNEWS to distribute a wire story to several queues when it is received. Likewise, two or more users can put copies of the same story into their personal queues. If you copy and distribute enough stories, a large portion of the database can become cluttered with the copies.

To avoid filling up the disk with copies of stories, your system keeps only the original story on the disk. When a story is copied to another queue, your system puts a reference to the original story in the queue that will hold the copy.

When someone opens a copy of a story, the system uses the reference to find the original story. It makes a working copy of that story, which it sends to the user. If the user examines the working copy and makes no changes, the working copy is deleted when the user closes the story.

However, if the user makes changes to the working copy and saves it, the system saves that copy as a story, replacing the reference.

In most cases, you can treat a reference to an actual story as if it were the story. The only time you need to take references into consideration is when setting purge intervals. When the system references a story to several different queues, each reference takes on the purge interval of the queue. When a reference becomes older than its queue's purge interval, the computer puts the reference, not the actual story, in the Dead queue.

The space used for a story becomes free only when the last reference to the story has been removed from the system.

# Tracking Database Space over Time

Do not wait until you encounter an "out-of-space" condition before you start to think about database storage. There are two ways you can get a good picture of space usage over time:

- The "hogs" report
- The information generated by the `dbfree` command

## Using the hogs Command to Obtain Information

The `hogs` command displays how much space particular queues are using in the database. It uses this format:

```
hogs  [<directory or queue name>]
```

To get a hogs report on the People directory, type:

```
hogs people
```

A screen similar to the following appears:

```
%   USED  SHARED HELD   LOCKED PURGE   QUEUE NAME

0   36    20     0      0      0       PEOPLE.LEVY.BYLINE
0   128   20     0      0      0       PEOPLE.LEVY.FINAL
0   32    40     0      0      0       PEOPLE.LEVY.FORM
...
0   425   40     0      0      0       PEOPLE.WALTERS.NOVEL
```

The `USED` column contains the number of blocks used in that queue. The `SHARED` column indicates the number of blocks used in that queue that are also used in other queues. To get the number of blocks used only by that queue, subtract the `SHARED` count from the `USED` count. If any queue has a substantially greater number of used blocks than the others, examine that queue more closely.

To obtain a hogs report on the entire database, type:

```
hogs .
```

To send a hogs report to a system printer, precede the `hogs` command with `print` and the number of the system printer you want to use. For instance, to send the report on the People directory to printer 3, type at the console:

```
print 3 hogs people
```

To send a hogs report to yourself, type:

```
NRCS-A# sh
# hogs . | mail <your username> &
```

*For more examples on how to use the* `hogs` *command, see Appendix A on page 138.*

## Using dbserver and dbfree to Obtain Information

**To obtain an accurate idea of how much space is being used by stories in the system:**

1. Empty out the Dead queue and reclaim all space used in it. Type:

   **`dbserver 16000000`**

*You must run* `dbserver` *when other programs like* `dbpurge` *are not running.*

2. The next day, or after completion, run the `dbfree` program to see how much space is being devoted to functions in your newsroom you consider critical.
3. Repeat this process from time to time, so you are aware of trends in space usage.

   Use the information to make decisions on projected storage needs, and how space is used.

# Increasing Database Space for Immediate Use

If your database has not reached the “Low on Space” point, but you want to increase free space for immediate use, do the following:

1. Type the `dbfree` command.

   ```
   NRCS-A$ dbfree
   data base size 4194304, free 224159 = 0%, freelist 769
   ```

The `dbfree` command—used by a system operator or a superuser—displays database size, free list size, and percentage of remaining database available. The `freelist` number indicates how many database blocks are used to hold the free list. The free list is compressed. This results in anywhere from 250 to 5,000 free-block

numbers contained in each free list block. In this example, 769 blocks are used to hold the 224,159 free block numbers. That is an average of 291 free block numbers per free list block.

2. Back up old material to tape and remove it from the database.
3. Check and reset purge intervals of Wires directories and queues.

   To view purge intervals at the console, type:

   **`list d wires`**

   Information similar to the following appears:

```
 SRPlo-LIsUG-QSXWFiT sortfield purge dis mbox directory
D-R-----I----Q-XW--- TITLE P4.0  D1  -  WIRES
D-R-----I----Q-XW--- TITLE P4.0  D1  -  WIRES.ALL
D-R-----I----Q-XW--- TITLE P4.0  D1  -  WIRES.AP
...
```

   One of the biggest consumers of database space in a newsroom is the Wires directory. Because wire stories lose much of their value after a few days, most newsrooms set a purge interval of 2 or 3 days. In this example, all Wires directories and queues have a purge interval of four days. The purge interval is a database trait you can customize for each database directory or queue. For more information on modifying database traits, including purge intervals, see Chapter 5 of the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

# Maintaining the Database

The following sections describe the two most important maintenance procedures you should perform regularly on your database:

- Check the database for errors
- Clean up the database on a monthly basis

## Checking the Database for Errors

For more information on the various commands used in this section, see Appendix A.

In the large databases typically maintained by newsrooms, minor errors can develop in some stories as a result of vast amounts of information the system processes every day. These errors can grow and eventually begin to damage the database if they are not removed.

To check stories in your database for errors, use the `dblines` console command, which examines every story in the database. You can run `dblines` on any server while the system is online. It can point out problems or errors in the structure of the database, find queue corruption that must be fixed before a `dbvisit` is run, and make some repairs. It cannot reclaim unreferenced blocks, adjust link counts, or rebuild the free list; `dbvisit` is required for that.

Since `dblines` examines every story in your database, it requires several hours to complete its task, so run `dblines` before you go home at night. By the time you return in the morning, it should be done.

When `dblines` discovers an error, it ordinarily sends an error message to the console. However, if you run it at night, you can have the system send its messages to a log file for you to review the next day.

**To execute the command and have it send output to a log file:**

1. Enter superuser mode on the master computer.
2. Type: **`dblines . > /tmp/dblines.log`**

You can use the `at` command to schedule the `dblines` command to run at a specified time. For instance:

1. Enter superuser mode on the master computer.
2. Get the current time, using the `date` command:
   ```
   NRCS-A# date
   Sat Jul 10 11:05:07 EDT 2004
   ```

You must use 24-hour military time.

3. Set the schedule, such as 5 minutes later, using the `at` command:
   ```
   NRCS-A# at 11:10
   ```
4. Type the `dblines` command line:
   ```
   /exc/dblines . > /tmp/dblines.log
   <Ctrl+D>
   ```

Pressing Ctrl+D after entering the command line marks the end of commands to be executed according to the `at` schedule. Afterwards, the server prompt returns to the original prompt, showing the server name, such as `NRCS-A#`.

If you want the system to fix script related errors, such as `story scripted flag set, story unscripted`, you can use the `+S` option. For instance:

```
/exc/dblines +S . > /tmp/dblines.log
```

If you want the system to ignore story script errors and not report them, you can use the `-S` option. For instance:

```
/exc/dblines -S . > /tmp/dblines.log
```

After the dblines is complete, you can check the log file for error messages, by using the `grep` command:

```
NRCS-A# grep failed /temp/dblines.log
```

This searches the log file for the word `failed`. If there are no errors involving queue corruption, such as `<queue> scan failed`, `<queue> read error`, or `unable to open <queue>`, the prompt will simply be returned. If there are any error messages related to queue corruption, call Avid Customer Support for assistance.

If you want to review the entire log, you can use the `doc` command to place a copy in a queue you specify. For instance, using the previous example, the sequence of commands (shown in bold) may appear similar to the following:

You must use 24-hour military time.

```
NRCS-A# at 11:10
/exc/dblines . > /tmp/dblines.log
doc -pu people.sysadmin.notes /tmp/dblines.log
<Ctrl+D>
jop 1001340600.a at Sat Jul 10 11:10:00 2004
NRCS-A#
```

In the example, the system would execute the `dblines` command at 11:10, send the results to a log file called `dblines.log`, and then place a copy of that file in the `PEOPLE.SYSADMIN.NOTES`. As indicated by the pound sign (#) in the prompt, you must be in superuser mode to execute the commands.

## Cleaning the Database

Over the course of a month, the iNEWS database may develop minor errors in its structure—the overall organization in which individual stories are arranged. These errors are the result of normal database cycles in which old material is removed and new information is added.

These errors grow and begin to damage the database if you do not remove them. The database cleanup procedure identifies errors so you can repair them before they become harmful.

*Unless you are instructed to perform this procedure more often, do a database cleanup once a month.*

### Cleaning Your Database Offline

The database cleanup procedure using the `dbvisit` command requires you to briefly take your system offline, preventing users from logging in.

**The day before performing the monthly database cleanup offline, do the following:**

1. Select the master computer (typically server A).
2. Type **`dbserver 16000000`** to empty the Dead queue.

Starting the day before, allows time for the system to complete this process, which may take several hours, before you run `dbvisit`.

*To check the database structure for errors,* `dbvisit` *must perform a cursory check of stories in the database, including those in the Dead queue. You can shorten the time it takes* `dbvisit` *to examine the database by using* `dbserver` *to empty the Dead queue first.*

*The* `dbserver` *command initiates a day-long operation and should be done one day ahead of the next steps in this procedure. Your system remains online during the operation.*

**After the `dbserver` operation is completed, you are ready to clean up the database offline, by doing the following:**

1. Shutdown the system, by completing the following steps:

   For more details on the procedure for shutting down the system, see Chapter 3 of the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

   a. Select *all* servers.

   b. Take the system offline.

   c. Log out all users.

      The `dbvisit` command cannot examine stories being edited. You must log out everyone on the system and stop all news programs on the servers before running `dbvisit`.

   d. Run `stop all` on *all* servers to stop all PCUs, workstations, wires, and other devices from making further changes to the database.

   After using the `stop all` console command, wait a few seconds for the prompt to reappear. When it does, proceed to the next step.

**Ensure system is offline and that you have stopped all PCUs and network devices. If you do not do this before cleaning your database, you may corrupt the data when you perform the cleanup.**

2. Start the database cleanup by doing the following:

   a. Select only the master computer (typically server A).

   b. Enter superuser mode.

c. Type:
**dbvisit -d**

A message similar to the following appears:
```
09:31:45 opening database
09:31:45 traversing roots
```

The `-d` instructs `dbvisit` to print a period each time it visits a new queue and a colon for every 1,000 stories in a queue. After you have done a few cleanups, you can judge how far `dbvisit` has progressed by the number of periods it has printed. When you run `dbvisit`, it begins with the root file structure and then examines the entire directory. As it inspects the root files and then the directory, `dbvisit` prints messages indicating its progress.

3. When you see the message `traversing directory`, do the following:

a. Select the master computer (typically server A) on your system.

b. Enter superuser mode—if you haven't already.

c. Type:
**dbvisit -dr**

*Running* `dbvisit` *on all servers instead of just the master computer is optional. If all servers are selected, the* `dbvisit` *command runs in read-only mode on the other servers, but running it on all of them allows you to compare their outputs and verify the database is mirrored on each server. The* `dbvisit` *command continues to run on the master computer.*

The dbvisit process has four main phases:

- Visiting the "root" blocks
- Visiting the directory and queue structures
- Visiting the stories
- Rebuilding the free list and fixing link counts

4. When the system prompts you with `Rebuild the free list (y/n)?`, do the following:

a. Ensure `dbvisit` has completed on *all* servers (otherwise error messages may appear).

b. Type **`y`** *only if there are no error messages*. If there are any error messages among the periods printed after `traversing directory`, do not rebuild the free list and contact Avid Customer Support immediately for instructions.

   -OR-

   Type **`n`** to skip rebuild of the free list.

   Skipping rebuild of the free list retains the old free list. While this means that unreferenced blocks are not collected back to the free list, you can add them to the free list later by performing another `dbvisit`.

*If there are link count discrepancies, the system will list them. Such advisory messages about high and low link counts are not errors, however, you will be prompted to correct them.*

**Rebuilding a free list without first fixing the errors seriously corrupts your database.**

5. If you do not see any error messages, select the master computer (typically server A) and type **`y`**.

   In addition to spotting errors in the database structure, `dbvisit` collects any unreferenced blocks and puts them on the free list. Unreferenced blocks are not used by any story and are not a part of the free list. They are stray blocks that cannot be used until they are put on the free list.

6. Restart your system, by doing the following:

   a. Select *all* servers.

   b. Exit superuser mode by pressing Ctrl+D.

   c. Type **`online`** to bring the system back online.

   d. Type **`restart all`** to restart all devices.

   The system displays `Hot-to-go` messages as the devices start. If a PCU or device cannot be started, a `failed` message is displayed for that PCU or device. Wait for the other devices to start, then use standard procedures to restart failed devices, as covered in the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

# Chapter 5

# Backing Up the iNEWS System

To properly maintain the iNEWS system, you need to perform three kinds of backups:

- Database backups—of the iNEWS database
- Software backups—of iNEWS software and the operating system
- System file backups—of files like /site/config and /etc/hosts

This chapter explains the procedures for the various types of backups required to properly maintain the iNEWS system. It also provides general information about using tapes and tape drives.

This chapter contains the following main sections:

- Tape Operations
- Backing up the iNEWS Database
- Restoring Data to the iNEWS Database
- Disaster Recovery Planning
- Backing up Software
- Backing up System Site Files

# Tape Operations

When making backup tapes, pay particular attention to the write-protect tab on the tape. If you have trouble writing to the tape, check the tab and try moving it in the opposite direction.

When loading a tape, ensure the tape has finished loading before issuing tape commands. Trying to access the tape before it is ready may result in a "hung" process that waits forever for the tape to be ready. The only solution is a reboot. For DAT tapes, wait at least a minute after all tape activity lights cease blinking to be sure the tape is properly loaded.

*Always clearly label tapes with the type of backup tape it is, date it was produced, and command used to produce it. Ensure that your tapes can be found quickly, in case of emergency.*

Since the database is mirrored on all servers in your system, it does not matter on which server you produce the database backup tape. You may want to rotate which server does the database tape dump so the tape drives wear evenly and you extend the life of the drives. Alternatively, you may want to always run the backup on one server's drive so you have another, relatively unused drive standing by in case the heavily-used drive develops problems. However, for software and site file backups, Avid recommends separate backups be made on each iNEWS Server.

Later in this chapter, various procedures are provided for checking data on tapes, searching for specific data on tapes, and restoring data from tapes. For more information, see "Listing Tape Contents and Backup Dates" on page 65 and "Searching a Tape" on page 68.

## Establishing Policies for Backup Procedures

Since your software and system files change infrequently, you do not need to back them up very often. Your database, however, changes hundreds or even thousands of times every day, so they need more frequent backups

Avid recommends your station establish and following a local backup policy for each type of data in your iNEWS system.

The following are some general backup guidelines:

- Since the database is mirrored across multiple servers you have built-in hardware redundancy. Backups also provide some protection against the rare cases of database corruption.
- One approach is weekly backups rotating through 5 tapes, one for each week of the month. If you need to keep old data for a period of time you may supplement the five tape rotation with 12 more tapes labeled with the months in the year.
- Do not store archived material beneath daily show production queues. Segregate archive material under it's own directory.
- If you archive old shows on your system, you may want to produce dbdump tapes that cover old years For instance, `ARCHIVE.2002` through `ARCHIVE.2003`, and store them separately.
- It is a good idea to make extra copies of the tapes and store them off site. If your computer room is destroyed, you can buy new servers. But if the computer room is destroyed with all backup tapes stored in the room, your data is lost. Many corporations use off-site storage services for backup tapes.

# Backing up the iNEWS Database

Database backups provide insurance against system calamities. Also, to free up space in the database, you can back up stories to a tape and then remove them from the database. Avid recommends making frequent backups of important material and the entire database. To back up database items to tape, use the `dbdump` console command.

## The dbdump Command

The `dbdump` command backs up the news database, including the People files, show rundowns, wire stories, and root database information such as user accounts, passwords, and directory traits. All user, database, and group traits are stored within the database.

The simplest form of the dbdump command is:

```
NRCS-A# dbdump c
```

The c stands for create. When you use dbdump c, it overwrites any information currently on the tape. The dbdump c does a full database backup of everything (except the contents of any queue or directory with its skip flag enabled.)

*The skip flag is a database trait that prevents the* dbdump c *command from backing up contents of queues and directories. Generally, Wires queues and the Dead queue are usually skip-flagged so they are not backed up. This results in less time for the backup and less tape used. If a dbdump reaches the end of the tape and still has more data to back up, you will be prompted for another tape. For more information on skip flags—also known as Skip Backup database trait—See the iNEWS Newsroom Computer System Setup and Configuration Manual for more information.*

For instance, to dump all database items contained in the SCRIPTS.2004 directory to tape, you could type:

```
dbdump c -n script.2004
```

If you already have dbdump material on tape and you want to append another dbdump to the end of it, use the following command:

```
NRCS-A# dbdump ad
```

An append does not overwrite dbdump information currently on the tape. See “Backing up Individual Queues” on page 60 for more information.

*To minimize the impact of any potential problems, perform* dbdump *at less critical usage periods.*

## Backing up the Entire Database

**To back up the entire database to tape:**

1. Insert a tape into a server’s tape drive.
2. Select the server that has the backup tape. For information on selecting a server, see Chapter 2 of the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

3. Type **`dbdump c`**.

   A verification request similar to the following appears:

```
Do you really want to create a new archive? (n/y)
```

**When backing up the database onto a tape, anything on that tape is overwritten by the new copy of the database.**

4. Type **`y`** to begin copying the database to tape.

   Information similar to the following appears:

```
Starting Dump(1), block(0)
dumping isam user
...
3121 stories dumped
Ending Dump(1), block(223)
```

   As `dbdump` copies, the console displays messages like those above. The number of stories dumped and the ending block mentioned in these messages depends on your system.

   When the console prompt returns, the backup is done. Proceed to step 5 to verify the backup. Otherwise, remove the tape from the drive. Set the tape's write protect switch to the on position, return it to its case, and write the date on the case label.

5. (Optional) Type **`dbrestore tdv`** at the console to verify the backup was complete—that everything you wanted backed up was copied to tape.

   The `dbrestore tdv` console command lists every directory and queue on the tape. Since the tape contains most of the database, this list is very long.

*To list contents of a tape, the server must read the entire tape; therefore, listing contents takes approximately as long as it does to back up the database to tape.*

The output of the `dbrestore tdv` command looks like this:

```
Listing tape contents only!
Type  Stories   Name
Dir
Que             DEAD
Dir             SYSTEM
Dir             SYSTEM.KEYBOARDS
Que   1         SYSTEM.KEYBOARDS.000-INSTALLATION
...
Que   3         TEST.SMITH
3630 stories listed
```

The first column of the listing identifies whether the item is a queue or a directory. If it is a queue, the second column indicates how many stories are in that queue. The third column displays the name of the directory or queue.

If any queues in the list do not have stories listed, then either the queue has no stories, or the queue has its skip flag enabled. If the skip flag is enabled, the queue name is copied to tape, but none of its stories are copied. In the previous example, the Dead queue does not have any stories on the tape.

6. When the list is complete, remove the tape.
7. Set the tape's write protect switch to the on position, and write the date on the case label.

## Backing up Individual Queues

Individual queues can be backed up by specifying the **`-n`** flag and listing up to 10 queue or directory names:

```
NRCS-A# dbdump c -n <queue or directory name>
```

**To back up individual queues or directories to tape:**

1. Insert tape into a server's drive and select that server.
2. Use the `dbdump` command in one of the following formats:

- If you do not have a tape that already contains a backup, insert a new tape and back up the queue. For instance, to back up SCRIPTS.2004.FEB on a new tape, type:

  **dbdump c -n scripts.2000.feb**

  A verification request appears:

```
Do you really want to create a new archive? (n/y)
```

  If there is nothing on the tape that you want to save, type **y** to continue. A message similar to the following will appear:

```
Starting track(1), block(0)
47 stories dumped
Ending track(1), block(34)
```

*When you back up a queue,* dbdump *ignores the queue's skip flag, so you can use it to back up a queue or directory that has its skip flag enabled. You can also use this process to save a queue or directory to tape before removing it from the database. For instance, if you want to restore database space by removing a queue with material no longer used, use* dbdump *to backup the queue first. Then, if you need it later, you can retrieve it from the tape.*

- If you have previous backups on the tape, append the next backup to the tape with:

  **dbdump a -n <queue or directory name>**

  You can append additional backups to that tape until you use up all space on the tape. Continuing with the previous example, the next day you would insert the same tape, make sure its write protect is off, and then use **dbdump a** to append that day's scripts to the tape:

  For instance, after making the first backup of SCRIPTS.2004.FEB shown in the previous example, back up SCRIPTS.2004.MAR the following month by inserting the same tape and entering the following:

  **dbdump a -n scripts.2004.mar**

*If you try to use* `dbdump a` *with a tape that does not already contain at least one backup, you get an Empty tape message. If this happens, use* `dbdump c` *instead.*

3. When the console prompt returns, the backup is done. Remove the tape and set the write-protect switch on. Date the tape case label.
4. To verify the backup, type:
   **`dbrestore tdv`**

   This list should be short, but it could take time to complete if the tape contains several weeks' worth of backups.

## Notes on Backing up the Database

If you try to back up data to a write-protected tape on the Linux platform, you get a message similar to the following:

```
NRCS-A# dbdump c
Do you really want to create a new archive? (y/n) y
Starting Dump(1) Block (0), dumped on Thur Jul15
‘0:36:14: 2004

dumping isam user
Please remove tape.
Insert continuation tape (make sure safety is off)
When continuation tape is ready type: y (RETURN)
```

If the tape is write-protected, remove it from the drive. On some tape drives you can change the write-protect without removing the tape. Set the write protect switch to off, insert the tape, and repeat `dbdump`.

Daily backups usually take very little time and generally do not affect system performance. However, backing up a large amount of material to tape may hinder system performance, so you should only do it during times of lighter system use.

If you have a very large database, the full database backup may require two or more tapes. The system prompts you to insert the next tape. Remove the current tape, number it as tape number one, and insert another tape. The server automatically continues copying the database when you insert the second tape.

# Restoring Data to the iNEWS Database

The following sections describe how to restore data from tape.

## The dbrestore Command

Tapes produced with the `dbdump` command can be read back in and restored using the `dbrestore` command. The most common use of the `dbrestore` command is to restore a specific queue or directory. This is done by specifying the name on the command line:

```
NRCS-A# dbrestore d -n <queuename> <queuename> ...
```

You can use the `dbrestore tdv` command to verify that you can properly read a `dbdump` tape.

Check free space before you restore large amounts of data to the database. Before you `dbrestore` stories or queues to your database from tape, ensure you have enough free space available to hold the restored stories. You can create free space using the `dbserver` command to empty out some of the Dead queue and add that space to the free list. A conservative estimate would be to have 10 free blocks for every story you plan to restore.

*When the* `-i` *option is used with the* `dbrestore` *command, dbrestore does not create any index requests in* `SYSTEM.INDEX` *when stories are restored to indexed queues.The dbrestore process is slowed when attempting to create index requests. Avid recommends the database be restored using the* `-i` *option and the stories be re-indexed at a later time.*

The `dbrestore` character mapping feature allows a single character to be mapped to one or two characters.

See "dbrestore Character Map (Arabic)" on page 268 for more information.

Map definitions come in three forms:

| | |
|---|---|
| 65 -> 97 | (a single to single map) |
| 65 - 90 -> 97 - 122 | (a range of single to single maps) |
| 247 => 225 194 | (a single to two-character map) |

*Notice that the difference of mapping to a single versus a double is an equal sign (=) instead of a dash (-).*

## Restoring a First-Level Directory

Restoring a first-level directory, such as Scripts, as opposed to `SCRIPTS.FEB`, is the same as restoring queues and other directories. However, when restoring such a directory, you also have the option of restoring it under a new name. Use:

**`dbrestore d -n <old dir. name>[=<new dir. name>]`**

If the new directory does not already exist, `dbrestore` creates it.

For instance, suppose you have a directory called Scripts and you want to restore a previous backup of this directory to another directory called Scripts-temp, which does not exist. Type:

**`dbrestore d -n scripts=scripts-temp`**

A message similar to the following appears:

```
744 stories restored
```

Since Scripts-temp does not exist, `dbrestore` creates it before restoring Scripts-temp. When `dbrestore` has restored all stories in Scripts-temp, it displays a message indicating how many stories were restored.

*If you want to restore a specific directory, be sure to always include the* `-n` *and the directory name. If you do not,* `dbrestore` *tries to restore everything on the tape to your database, which could cause your system to run out of space.*

You may not restore a first-level directory to a lower level. In the example above, for instance, we could not have restored the Scripts directory to `SCRIPTS.MAY`.

When you restore a first-level directory to an existing directory, it acquires the traits of that directory. If `dbrestore` must create the directory, it uses database traits of the directory it is restoring.

**If the new directory name is longer than the old one, there is a possibility that sub-directory names may become too long and cannot be restored. When the new name is longer than the old name, the `dbrestore` program will prompt with the following warning message: `WARNING: Alias too long!`**

**It should be shorter or the same length as current directory name to avoid exceeding directory name length restrictions.**
**`Do you really want to continue? (n/y)`**

## Listing Tape Contents and Backup Dates

When restoring a directory, you may need to examine the tape's contents to ensure the directory you want to restore is actually on that tape. Otherwise, if you attempt to restore a directory or queue that is not on the tape, the system may take several minutes to search the entire tape before reporting `no stories restored`.

The pages that follow show you how to use the `dbrestore` command to list the directories and queues that have been backed up on a tape, to find out if the tape contains any stories saved after a certain date, and to find out the date of each backup on the tape.

### Listing Contents of a Tape

To list every directory and queue on the tape, as well as the number of stories in each queue, to the server with the tape drive, type:

**`dbrestore tdv`**

Information similar to the following appears:

```
Listing tape contents only!
Type  Stories   Name

Dir
Que             DEAD
Dir             SYSTEM
Dir             SYSTEM.KEYBOARDS
...
Que   3         TEST.SMITH
3630 stories listed
```

Each line in the list consists of three columns and identifies a directory or a queue that has been saved to the tape:

- The `Type` column indicates whether the line lists information about a directory or a queue. Directories are identified by `Dir` and queues by `Que`.
- The `Stories` column is used only in lines beginning with `Que` and identifies how many stories are in that queue. Use this number as a guide to how much space you need to restore the queue. If there is no number, the queue is empty. (The queue is empty because either no stories existed in the queue or the queue's skip flag was enabled.)

The `Name` column identifies the directory or queue by name.

If you attempt to restore a queue with no stories, you see the following message:

```
no stories restored
```

### Listing Items Dumped on a Particular Date

**To list items dumped on a particular date:**

1. Use the following format of `dbrestore`:
   **`dbrestore td -d <date>`**
2. Enter the date in `YYMMDD` (year, month, day) or `YYYYMMDD` format.

Begin single-digit months and days with a zero, such as `04` for April. Also check for items dumped to tape during a month by using the year and the month (`YYMM`).

You can check to see what was dumped to a tape during a year by specifying just the year (`YY`).

For instance, to list everything dumped in October 2000, type:

**`dbrestore td -d 0010`**

A message similar to the following appears:

```
Listing tape contents only!
Type  Stories   Name
Dir
Dir             SHOW.TODAY
Que   34        SHOW.TODAY.SCRIPTS
Que   3         SHOW.TODAY.RUNDOWNS
Continuing with  Dump(2)  Block(0)
Dumped Thu Jul15 10:30:19 2004
Dir             SHOW.TODAY
Que   27        SHOW.TODAY.SCRIPTS
Que   3         SHOW.TODAY.RUNDOWNS
67 stories listed
```

In the previous example, the tape contains only backups from two dates: the first backup and one made on July 15, 2004. Notice the date of the first backup is not included in the listing. To find the date of the first backup, use the `dbrestore tf` command described below.

The `dbrestore` displays a `Continuing with Dump` message just before processing the second dump.

### Listing the Date of Each Backup

To list the dates of each backup on a tape, type:

**`dbrestore tf`**

A message similar to the following appears:

```
Listing tape contents only!
Starting:  Dump(1)  Block(0)
Dumped Tue Jul10 15:33:01 2004
Continuing with Dump(2)  Block(0)
Dumped Thu Jul15 10:30:19 2004
Elapsed time: 00:13
```

In the example, the tape contains two backups, one done on July 10, 2004, and another done on July 15, 2004.

## Searching a Tape

You can retrieve individual stories from a tape by searching for a word contained in the story.

### The searchtape Command

The `searchtape` command allows you to search through a tape looking for a specific word. Stories that contain that word are restored to the `SYSTEM.SEARCHTAPE` queue. There is a maximum number of stories or hits that will be restored with the `searchtape` command. This default is specified in the `/site/system` file. See "Searching a Tape by Word(s)" on page 69 for more information.

*When you list a tape's contents, your system must read all information on the tape to generate a list of the contents. If the tape contains much information, listing the tape's contents takes a significant amount of time.*

*Because searching tapes for stories takes a long time and can reduce system performance, restrict tape searches to periods of lighter system use. Avid recommends running all tape operations during non-critical periods, not during shows.*

### Searching a Tape for Stories

**To search a tape for stories:**

1. Combine one or more keywords with a date or range of dates, as described in the following sections.
2. When the system finds a story matching the search criteria, it restores the story to the database. Because stories on tape may be old versions of stories still in the database, the system restores them to the `SYSTEM.SEARCHTAPE` queue. This prevents the system from putting an old version of a story in a queue that contains the current version of the same story.

After restoring a story to `SYSTEM.SEARCHTAPE`, you can move it to any queue.

*Before restoring stories from tape, check how much free space remains in your database at the console. See "Checking for Free Space on a Database" on page 72 for more information.*

### Searching a Tape by Word(s)

To search a tape by word(s), follow the `searchtape` command with the word `for` and the word(s) for which you want to search.

Use this form of the command:

```
searchtape  for  <words>
```

If you include more than one search word, the system searches for stories that contain any words for which you are searching. You can search for up to 20 words at a time with this command.

For instance, to search for stories that contain either helicopter or president, type:

```
searchtape for helicopter president
```

A message similar to the following appears:

```
8 stories restored to SYSTEM.SEARCHTAPE
```

This command finds stories with the word president and it also finds stories with the word helicopter. It would not limit the results to stories that contained *both* president and helicopter.

*It is not possible to construct Boolean (*and) *searches using the searchtape command.*

### Searching a Tape by Word and Date Range

**To restrict a search to stories backed up between certain dates:**

1. Include the range of dates to which you want to restrict the search in the `searchtape` command.

   The command format for this kind of search is:

   ```
   searchtape from <date> to <date> for <words>
   ```

Dates must be in `YYMMDD` (year, month, day) or `YYYYMMDD` format, and single digit months and days must begin with a zero (for instance, April is `04`). If you want to specify only a month or year, shorten the date format to `YYMM` or just `YY`.

2. Follow the second date with the word `for` and the word(s) for which you want to search.

   To restore stories archived between May 1, 2000, and May 21, 2000, that contain the word helicopter, type:

   ```
   searchtape from 000501 to 000521 for helicopter
   ```

   A message similar to the following appears:

```
6 stories restored to SYSTEM.SEARCHTAPE
```

*The* `searchtape` *command looks for stories saved to tape within the dates you specify. It does not look for stories created within those dates.*

### Searching a Tape by Word and Day

To search for stories saved to tape on a particular day, use the search format:

```
searchtape just <date> for <words>
```

To search for stories backed up on May 15, 2000 that contains the words helicopter or president, type:

```
searchtap just 000515 for helicopter president
```

### Searching a Tape by Word and Month

To search for stories saved to tape during a particular month, follow `searchtape` with `just` and shorten the date to the year and month (`YYMM`).

For instance, to search a tape for any story saved to tape in February, 2000 that contains helicopter, type:

```
searchtape just 0002 for helicopter
```

### Specifying a Maximum Number of Stories to Search

To specify a maximum number of stories for `searchtape` to restore, include the word `max` and the number of stories you want to restore.

To search a tape for all stories that contain the word president, type:

```
searchtape for president
```

If you do not specify a maximum, the system restores up to 50 stories.

If you specify a maximum number, `searchtape` stops when that number of stories has been restored, or when all stories containing the text have been found—whichever comes first.

To search a tape for the first 10 stories containing president, type:

```
searchtape max 10 for president
```

Anytime you specify more than 50 stories to restore, you are reminded that this could cause the system to run out of disk space.

When the search is complete, `searchtape` stops and displays a message indicating the number of stories it has restored.

To cancel the restore operation at any time while in progress, press Delete.

## Checking for Free Space on a Database

### To check free space on a database:

For information on selecting a server, see Chapter 2 of the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

1. Select a server.
2. Type:
   **`dbfree`**

   A message similar to the following appears:

```
data base size 4194304, free 224159 = 5%, freelist 769
```

In the previous message example the `freelist` item indicates how many database blocks are used to hold the free list. The `freelist` is compressed. You should have 10 free blocks for each story you want to restore. For instance, if you expect to find 10 stories to restore, you should have 100 free blocks. Add extra space to the free list before conducting a search because the search may find and restore more stories than you expect.

### Adding Blocks to the Free List

### To add 100 blocks to the free list:

1. Select the master computer (typically server A).
2. Type:
   **`dbserver 224259`**

   The number after `dbserver` must be the sum of the additional blocks and the free blocks from `dbfree`.

   A message similar to the following appears:

```
NRCS-A Sat Jul24 10:29:26 2004 dbserver complete-exiting
```

At this message, `dbserver` is finished and you may proceed. You can run `dbfree` again to verify that you have enough space.

## Restoring the Database

When a story being restored has more than 1,000 lines, the following message is displayed:

```
Warning: Large story being restored.
     Queue: FUTURES.FEBRUARY
     Title:  "Senate Hearing"
     Size:  1004 lines (approximate)
     Press <del> to skip this story.
```

If the story size continues to grow, the following message prints every 1500 lines:

```
Large story continuing: 1500 lines
Press <del> to skip this story.
```

To skip the large story, press Delete while it is being restored.

Always include `for` and one or more words for which you want to search when you use `searchtape`. Even if you use a date or a date range as part of the search criteria, you must include a word or words for which you want the system to search.

Use search criteria as specific as possible:

- Always use a narrow range of dates
- Use a small number of search words
- Use words likely to appear only in stories for which you are searching

# Disaster Recovery Planning

Having a complete `dbdump` of the database ensures that you can restore your data in case of calamity; however, restoring a complete `dbdump` to a freshly initialized database can be very time-consuming and leave the

newsroom without access to the system until the full database `dbrestore` finishes. On very large databases, the full database `dbrestore` can take from 12 to 36 hours.

Planning a separate, efficient disaster recovery `dbdump`/`dbrestore` procedure can significantly speed up the time before the system becomes functional to users.

## Disaster Recovery Dbdump

The first step is to create a stripped down `dbdump` that contains the directory structure and minimal amount of queues and directories needed to bring the system online. After the minimal `dbdump` was restored, users can start working while the remainder of the database is restored from the last full database `dbdump` tape.

### Create Minimal dbdump

The entire structure of the database, along with your designated minimal set of queues, can be backed up, specifying the list of essential queues (or directories) to be backed up:

```
NRCS-A# dbdump ci -d -n <directory or queuename> ..."
```

*The* `-d` *option dumps the news directory skeleton but will not dump any stories. The* `-n` *option will dump the stories in any directories and/or queues specified. The* `i` *option will dump the isam files.*

The essential directories would be System and possibly Show and a few daily Assignments queues. The System directory must be part of the minimal `dbdump`.

If master rundown skeletons are stored elsewhere, add them to your list of essential queues. This may be the bare minimum of directories needed to get a show on the air.

*Do not store archives of shows beneath the Shows directory. Having years of archive material mixed in along with your rundowns and master rundowns will slow the* `dbrestore` *significantly. If restoring Assignments queues, ensure you restore only daily queues needed to operate; do not include all future files in the minimal* `dbdump`.

You may list up to 10 directories and queues with the `-n` option of the `dbdump` program.

```
NRCS-A# dbdump ci -d -n system shows assignments
```

In case of disaster, this minimal `dbdump` would be the first tape restored to the system. Since it will not contain many stories, to `dbdump` to or `dbrestore` from this tape is relatively quick and users can resume work while the remainder of the material trickles in from the full `dbdump` tape.

This minimal `dbdump` tape should be updated periodically.

# Backing up Software

Your system uses programs and site-dependent configuration files, called site files, to adapt it to your newsroom operations and run the workstations, printers, wires, and other devices. These programs and site files make your system software unique from that found in any other newsroom.

It is important you keep a current copy of your software on tape in case you need to restore the software area of the disk. Anytime you make major changes to the system software—such as extracting new programs from an update tape—make a new software backup.

Likewise, when you make major changes to a site file, update the software backup to include these changes. For instance, if you add a new workstation to your system, you define that workstation in the configuration file. This represents an important change to this site file so, after modifying this file, update your software backup tape.

# Backing up System Site Files

Your system uses site files in combination with software taken from your release CD and update tapes. When you do a software backup, both your system software and site files are copied to tape.

If something happens to your software backup, you may recover your programs from the release CD, but your site files could be lost. In addition to backing up the software, back up your site files on a separate tape. If anything happens to the software backup, you can use your release CD and site file backup to rebuild your system software.

Make a new site file backup after making significant changes to any site file. For instance, adding a printer to your system requires that you modify your configuration file (`/site/config`) and create a new printer profile. After making these changes, make a new software backup and then make a new site file backup.

To make a separate `sitedump` tape for each server, repeat this procedure on each server. After backing up a server's site files, ensure you label the tape to clearly indicate which server's site files are on it.

## The sitedump Command

The `sitedump` command backs up files in the `/site` directory and certain key files in the `/etc` directory. This is where all of your site-specific configuration files are kept. The tape should be updated whenever you make changes to your configuration.

**To back up your system's site files to tape:**

1. Insert the tape into the master computer (typically server A).
2. Select the master computer.
3. Enter superuser mode.
4. Type:
   **`sitedump`**

   This takes a few minutes, and no messages are displayed.

Linux systems can make sitedumps to their floppy drives. The floppy diskette does not need to be formatted first. Insert a 1.44 MB diskette in the system, and type:

```
sitedump -f /dev/fd0
```

5. When the `sitedump` is done and the prompt returns, exit from superuser mode by pressing Ctrl+D.
6. Remove the tape and label it. Include the date and type of backup.

# Chapter 6

# Disconnects

This chapter provides vital information to help you recover from a system disconnect.

This chapter contains the following main sections:

- Normal System Status
- Disconnects
- Disconnect Recovery

# Normal System Status

The displays will vary when the system is a triple system configuration.

On a dual server system, the status command will show the system is AB, with both servers connected to each other. A display similar to the following will appear on server A:

```
NRCS-A#  status
A is ONLINE and has been CONFIGURED.  ID is NRCS.
System is AB.  Master is A.
Disk status is OK. The database is OPEN.
```

The system status is reported identically on both servers in the system. For instance, a display similar to the following will appear on server B:

```
NRCS-B#  status
B is ONLINE and has been CONFIGURED.  ID is NRCS.
System is AB.  Master is A.
Disk status is OK. The database is OPEN.
```

When the system is dual and databases are mirroring between the two servers, a story written on one server is automatically mirrored to the database of the other server. When the system is connected and has the normal prompt, the servers are in communication with each other and the disk mirroring process is active.

# Disconnects

If the servers disconnect from each other, the link is severed and stories written on one are no longer mirrored to the other server.

After the system has disconnected, the databases are no longer mirrored between the two servers and immediate attention is required. Since mirroring has stopped, stories written by users on server A will not be seen by users logged in to server B. There will be two entirely separate databases.

There is no way to reintegrate two disparate databases. You must decide which database will be retained as the master database, and the other database must be erased. The steps for recovering from a disconnect are outlined later in “Procedures” on page 85.

If a disconnect occurs, users may want to export the stories they are working on to local hard drives, in addition to normally saving them to the server. They can export stories by selecting File > Export Story. This will make a backup copy of the story on the local hard drive of their PC. Since they may be logged in to the server containing the non-master database that will be erased, this will give them a backup copy, which could be re-imported if necessary.

**When the servers disconnect, stories written on server A are not mirrored to server B and stories saved on B are not mirrored to A. The system issues a popup to alert users who must contact their system administrator immediately!**

## Detecting a Disconnect

You can check to see if the servers are connected to each other and mirroring at any time using the `status` command at the console. If they are connected and properly mirroring, they will both agree on the system status and it will report that the system is dual (AB):

```
System is AB.
```

See “Types of Disconnect” on page 82 for more information.

If the servers disconnect, a disconnect warning message appears on the console. Similar warnings appear on the console for other servers in the system of each server.

On the Linux platform, the messages include detailed information from the driver (mp) that controls the mirroring:

```
Jul 13 07:45:44 nrcs-a IO handler: B silent for 30 seconds
Jul 13 07:45:44 nrcs-a IO handler: LINK TO B FAILED, DISCONNECTING B
Jul 13 07:46:05 nrcs-a msg:   66 to server on computer A
Jul 13 07:46:25 nrcs-a S508: [15028] monitor 508  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S510: [15029] monitor 510  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S238: [14998] SEEK 238  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S240: [14999] SEEK 240  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S236: [14997] SEEK 236  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S232: [14995] SEEK 232  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S234: [14996] SEEK 234  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S254: [15001] ACTION 254  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S504: [15026] monitor 504  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S506: [15027] monitor 506  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S508: [15028] monitor 508  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S510: [15029] monitor 510  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S238: [14998] SEEK 238  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S240: [14999] SEEK 240  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S236: [14997] SEEK 236  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S232: [14995] SEEK 232  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S234: [14996] SEEK 234  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S254: [15001] ACTION 254  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S258: [15003] ACTION 258  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S252: [15000] ACTION 252  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a S256: [15002] ACTION 256  0 Server -  Hot-to-go
Jul 13 07:46:25 nrcs-a disconnect: B COMPUTER DISCONNECTED
```

In addition to console warning messages, a warning message is broadcast to all users currently logged in to the system. This warning appears on the status bar (if displayed) at the bottom of the iNEWS session.

## Types of Disconnect

When the servers disconnect, one may disconnect from the other, or they may both disconnect from each other.

If they have disconnected from each other, each will report that it is a single system, with itself as the master:

```
NRCS-A#  status
A is ONLINE and has been CONFIGURED.  ID is NRCS.
System is A.  Master is A.
Disk status is OK. The database is OPEN.
```

The single system status is reported identically on both servers:

```
NRCS-B#  status
B is ONLINE and has been CONFIGURED.  ID is NRCS.
System is B.  Master is B.
Disk status is OK. The database is OPEN.
```

The other possibility is that servers will disconnect, but one of the servers will not note the disconnect. In this case, one will report that the system is single while the other states the system is still AB.

Regardless of the report, once one of the servers has disconnected, the system must be recovered following the procedures in this chapter. The steps outlined are the only way to recover and get the servers back in mirror.

**If the system has disconnected, you cannot simply reboot the servers and bring it back up normally. Rebooting and connecting two servers together after a disconnect can lead to database corruption!**

## Causes of Disconnects

Servers are normally in constant communication with each other. When a story is saved, the server tries to mirror that change across to the other server's database. If the server cannot contact the other server for a period of 30 seconds, it assumes the worst—that the other server has died and is not available and that as the surviving server it must be responsible for the entire system.

Knowing this design, it is obvious that network outages will cause a disconnect, as will the loss of power by one server.

A “dirty” network leading to numerous network output errors (called `Oerrs,` as revealed by the `netstat -i` command) can cause a disconnect, particularly if the output errors are rapidly climbing.

A software error that leads to a looping condition that causes a server to become so busy it cannot respond to a mirroring request could also theoretically lead to a disconnect.

Hardware failures such as the failure of a network card or hard drive may also lead to disconnects.

# Disconnect Recovery

This section provides an overview of recovering your system from a disconnect, recovery procedures, and a quick reference worksheet you can use should a disconnect occur.

## Overview

After a system has disconnected, one server must be selected to continue on as the master computer. This server will be referred to as the *survivor.* The other server will be referred to as the *failed server.*

Before the failed server can be reconnected to the survivor, it must be rebooted and its database wiped clean. After the database on the failed server has been cleared, the server can be reconnected to the survivor and the master database copied back across from the survivor.

Because one server’s database will be selected as the master database and the other’s database erased, discovering a disconnect as soon as possible minimizes the possibility of data loss.

In normal dual-server operation, half the devices and sessions are configured on one server and the other half are configured on the other server. The most important thing to do after a disconnect is to reconfigure

the survivor so that it knows it must be responsible for *all* devices and sessions. You can then restart any network devices that were running on the failed server.

The steps are covered in more detail in the next section, “Procedures” on page 85.

**The disconnect recovery steps are as follows:**

1. Reconfigure the system (survivor).
2. Restart the failed servers’s network devices (PCU/MCSPCs) on the survivor.
3. Reboot the failed server.
4. Clear the database on the failed server.
5. Reconnect the failed server.
6. Copy the database to the failed server.
7. Log out and stop the failed server’s transferred network devices, sessions, and servers running on the survivor.
8. Reconfigure the system.
9. Start up the transferred network devices and servers back on the failed server.

## Procedures

This section contains an example of recovery from a disconnect of a dual AB system. The steps may vary when the system is a triple system configuration. Triple system configuration customers may need to contact Avid Customer Support for more specific instructions.

In the example, it is assumed that server A was chosen as the survivor and server B was designated the failed server:

NRCS-A – survivor

NRCS-B – failed server

The failed server is also referred to as the *revived server* once it is reconnected to the system.

*If you reverse the roles of the servers during your disconnect and make B the survivor and server A the failed server, remember to adjust which server you issue the commands on appropriately! The steps shown in the following example assume server A is the survivor; reverse A and B throughout the process if server B is chosen as your survivor.*

It is assumed (for this example) that when the servers disconnected, server A became a single system (`system is A, master is A`) and server B also became a single system (`system is B, master is B`). In this situation, users logged in on A can continue working and saving stories. Users logged in on B can also continue working and saving stories, but since the servers are not mirroring, anything saved to server A's database is not copied over to server B's database, and vice versa.

The databases on the machines are no longer mirrored. The only recourse is to choose one of the servers' databases to become the master database. The database on the other server is wiped clean and then recopied from the master server.

To export a story to a local hard drive, click File drop-down menu and select Export Story.

Users logged on to the failed server (B) are creating stories on a database that is going to be wiped clean. That information will be lost unless stories created after the disconnect are first exported to the local hard drive so they can be imported to the survivor once the user logs in to the other server.

The following are steps necessary to recover the system:

### Step 1: Choose one master machine to continue working with (survivor)

It does not matter whether you choose server A or B as your master computer. What is important is to choose one quickly. You may choose one server over the other if it has more users logged in on it. Or you may choose the server that has the show producer logged in on it. If you are about to enter a show, you may pick the one that runs the teleprompter. Or you may choose A as the survivor so the steps exactly shadow instructions in this section.

### Step 2: Reconfigure the master computer (survivor)

When you run the `configure` command, the master computer looks at the current system configuration and then consults the appropriate host section of the configuration file (`/site/config`) to see devices for which it is responsible.

If the system is A, then the configuration section for `host a a` is consulted. This section will contain all the device and session numbers, including those that normally run on the failed server (B).

For more information on superuser mode, selecting servers, and configuring the system, see the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

You must enter superuser mode at the console to configure the system:

```
NRCS-A$ su
Password:
NRCS-A#
```

and then run this command sequence:

```
NRCS-A#  offline
NRCS-A#  configure
NRCS-A#  online
```

### Step 3: Halt and reboot the disconnected (failed) server.

Type the following commands:

```
NRCS-B# /sbin/shutdown -f -r now
```

### Step 4: Reboot the disconnected (failed) server's network devices (PCUs and MCSPCs).

It may not be necessary in all cases to reboot the PCUs and MCSPCs. They may restart normally without a reboot. Reinitializing them by cycling the power virtually guarantees successful restarts.

*Keep a current printout of your* `/site/config` *file near your control console for handy reference during emergencies such as disconnects.*

### Step 5: Restart the disconnected (failed) server's network devices (PCU/MCSPCs) on the survivor.

Restart the failed server's PCUs and MCSPCs on the survivor:

```
NRCS-A#  restart <number>
```

You may string multiple device numbers on one line, like this:

```
restart 10 20 30 19 145
```

After you restart all failed (disconnected) server's network devices on the survivor, the survivor will be running all devices in the system. It is not necessary to restart the failed server's utility programs, because they are automatically started on the survivor when the disconnect is detected.

### Step 6: Log in to the rebooted failed server as system operator, then become superuser.

At the login prompt, log in as the system operator user (so):

```
iNEWS Newsroom Computer System
login: so
Password:
Last login: Mon Jul 19 07:17:23 on ttyS0
?:
```

When you log back in you will be at the question mark prompt because the server is not yet named (connected). You must be in superuser mode for the next step:

```
?:su
Password:
?#
```

### Step 7: Wipe clean the disconnected (failed) server's database

Select the failed (disconnected) server on the console and use the `diskclear` command to wipe the database off the failed server. The display will look similar to the following (with what you type appearing in bold):

```
?# diskclear -
DANGER -- This program DESTROYS all information on this
computer's data base.
Do you wish to save the current data base? (y/n): n
Are you sure you wish to CLEAR the disk? (n/y): y


?  Mon Jan  3 16:18:23 2000 diskclear CLEARING DATA BASE
?  Mon Jan  3 16:18:23 2000 Each dot represents 10,000
blocks. The entire database = 1677 dots.
........10.........20.........30.........40.........50
.........60.........70.........80.........90.........100
.........110......
```

The `diskclear` will print a sequence of dots and numbers as it clears the disk. For instance, on a full 16 gigabyte database, the dots and numbers count to 16,000. The `diskclear` may take some time to complete.

### Step 8: Reconnect the failed server to the survivor.

After the `diskclear` has completed you can reconnect the servers. Select both (*all*) servers on the console and type:

```
reconnect <failed> master=<survivor> net=ab
```

Depending on which one failed and which one is master, you would enter one of the following commands:

**`reconnect a master=b net=ab`**

-OR-

**`reconnect b master=a net=ab`**

A few moments after entering the command, the failed server will regain its normal, named prompt. Communication and mirroring between the servers is reestablished.

### Step 9: Begin copying the database back over to the failed server.

Select the revived (failed) server only on the console and begin a `diskcopy`:

```
NRCS-B#  diskcopy
```

Users can continue working while the `diskcopy` continues in the background.

The following steps can be performed at any time after the reconnect.

### Step 10: Stop the revived server's network devices (PCUs and MCSPCs), utility programs (such as action servers), and sessions on the survivor.

At this point, all devices and sessions are running on the survivor—that is, master computer—even the ones that normally run on the other, revived server. These devices, utility programs, and sessions eventually need to run in their normal place on the revived server.

Two methods are available for splitting the workload and putting things back in their normal places. The first method involves logging off and stopping half of the system. The second method is easier and involves briefly logging off all users and stopping all devices.

Method 1

If you cannot afford to log all users off for 10 minutes or so, the first step to transferring them back to their rightful place is to determine which devices and sessions belong on the revived server. Those devices and sessions must be logged out and then stopped. Utility programs (such as action servers and txnet links), PCUs and MCSPCs need not be logged out; they can be stopped directly.

```
NRCS-A#  stop <number>
```

Consult the appropriate host section of the `/site/config` file to see which devices must be transferred back to their normal locations. Before stopping them, take the system offline to prevent new logins, then type

**`list s`** to see whether users are logged in on the session numbers that must be stopped. These users must log out for a few minutes. Log them out, then stop the devices and servers:

```
NRCS-A$ offline
NRCS-A$ list s 10
P11                         A
P12                         A
T15          palmer         A
D16                         A
NRCS-A$ logout 10
NRCS-A$ stop 10
```

Only iNEWS Workstation sessions need to be logged out; they need not be restarted since users initiate the sessions when they login.

You may need to consult your printout of the `/site/config` file to determine which server and session numbers must be stopped. They are the numbers in the `host ab b` section. Ensure that you stop *all* of the failed server's utility programs—action, txnet, rxnet, and special servers—that are currently running on the survivor.

Method 2

On larger systems, logging off and stopping half of the devices, utility programs (such as action servers), and sessions is sometimes more difficult and time consuming than simply logging everyone off and stopping all devices. Rather than hunting through the `/site/config` file and determining which sessions and devices belong to the failed server, it may be quicker to schedule a time to bump everyone off.

Alternatively, if you have the opportunity to schedule a time to log the users off and stop the devices, you can opt for the second method and instead run the following sequence of commands on the survivor:

```
NRCS-A#  offline
NRCS-A#  logout all
NRCS-A#  stop all
```

### Step 11: Reboot the PCUs/MCSPCs that you have stopped.

Rebooting the network devices before getting them restarted ensures the devices come up cleanly.

### Step 12: Reconfigure the system (survivor).

The system is now running in a dual AB configuration. When you run the configure command, the system will reconsult the `/site/config` file and divide responsibility for which server will run the devices, half for server A and half for B:

```
NRCS-A#  offline
NRCS-A#  configure
NRCS-A#  online
```

Wait for the system being configured messages to appear on both servers before moving on to the next step.

### Step 13: Start up the revived server.

When you run `startup` on the revived server, the devices and utility programs (such as action servers or txnet links) on that server are started and it is placed online:

```
NRCS-B#  startup
```

### Step 14: Put the master computer back online.

The system is now running normally with A processes and devices running on server A, and B processes and devices running on server B. The `diskcopy` command will continue to run in the background, working to mirror the entire database. The `diskcopy` process will send progress messages to the console screen and eventually report that the `diskcopy` is complete. You can check whether it has completed by running the `status` command. The disk status will be `UNKNOWN` until `diskcopy` completes; at that time it changes to `OK`.

### Step 15: (If you used the alternate method 2 in step 10 and typed logout all and stop all)

If you decided to log out all users and stop all devices in step 10, you should restart all of the survivor's devices by typing **`restart all`** on the master computer. The system is now running normally (dual) with all sessions and devices in their normal places. The `diskcopy` may still be running in the background, copying the database back to the failed server.

If server B was selected as the master database, it is now the master computer. Since either server can run as master, nothing further needs to be done. If you wish to make server A the master computer again, wait until the `diskcopy` completes and then perform a normal system shutdown, reboot and startup, as described in *iNEWS Newsroom Computer System Setup and Configuration Manual*.

**Wait for the `diskcopy` to complete before rebooting the system to make server A the master computer again. Database corruption may result if the system is taken down before the *`diskcopy`* completes. If the system loses power or must be rebooted; the master computer that contains the full database must be brought up in a single server configuration, and the disconnect recovery procedure restarted again.**

## Recovery Worksheet

The following Disconnect Recovery table lists:

- Commands involved with recovering from a disconnect
- Whether the command is run on the failed server, survivor, or both servers
- Offers a column where you can fill in the name (either A or B) of the server that fits that role (either survivor or failed)

You may want to photocopy this table and write in names (letters) of your survivor and failed servers in the boxes for each step.

Before proceeding, review and understand the narrative steps in the previous section "Procedures" on page 85.

The table's procedure shows Method 2 (`logout all`, `stop all`) of splitting the workload to get the devices, utility programs, and sessions back to their regular positions.

| Computer | A/B | Commands to Run: |
|---|---|---|
| survivor | ❑ | Enter superuser mode.<br>**offline**<br>**configure**<br>**online**<br>Reboot failed server's network devices.<br>Restart failed server's network devices. |
| failed | ❑ | **/sbin/shutdown -f -r now**<br>Login as system operator.<br>Enter superuser mode.<br>**diskclear -** |
| both | AB | `reconnect <failed> master=<survivor> net=ab`<br>Type either:<br>**reconnect a master=b net=ab**<br>-OR-<br>**reconnect b master=a net=ab** |
| failed | ❑ | **diskcopy -s** |
| survivor | ❑ | **offline**<br>**logout all**<br>**stop all**<br>**configure** |
| both | AB | *Wait for the "system being configured" messages on both* |
| survivor | ❑ | **restart all**<br>**online** |
| failed | ❑ | **startup**<br>*Be patient for the server to come up.* |

# Chapter 7

# Troubleshooting

This chapter provides information to help you recover from various kinds of system failures.

This chapter contains the following main sections:

- iNEWS Workstation Problems
- PCU Wire Ports
- Locked Blocks
- How to Check Process Status (ps Command)
- Power Failure
- If they are plugged into different UPSs and they ran through the UPS battery and then lost power and rebooted, you will not know if they went down concurrently. One UPS may have run longer than another. If so, more stories may have flowed in on a wire or been saved by a user while the one server was still up and the other was down. The database will not be in mirror and you will need to go through the recovery process as noted in Chapter 1.
- Network Failure
- iNEWS and FTS Connectivity (ftsdebug Command)

# iNEWS Workstation Problems

This section explains common problems users may have during daily operation. Possible solutions appear after each problem.

*A user who installs the client software, iNEWS Workstation, on a computer running a Windows-based operating system must have permission to overwrite system files necessary to run the product (as in* `MFC42.DLL`*). Log in as administrator and set privileges on user accounts to enable access and modification rights on the system32 directory.*

## A User Cannot Log in

If a user is unable to log in to the iNEWS Workstation and gets the `Invalid User Name/Password` error message, ensure the user:

- Entered correct user name and password, spelled correctly
- Specified proper server name

Use the `list u` console command to verify a user with that name exists on the system.

If the password is a problem, assign the user a new password. Then, force a password change to maintain security. See Chapter 3 for more information.

## A User Cannot Establish a Session

If the user is properly entering user name, password and server name, but the login attempt "hangs" and then begins to cycle through alternate server names, wait until the login attempt times out and follow these steps:

1. Try to `ping` the server to check network connection to the server, by doing the following.
   a. Click Start.

b. Select Programs > MSDOS Prompt to open a DOS window. (The name and location of this submenu option may vary depending on the Windows-based operating system. For instance, it may appear at Programs > Accessories > Command Prompt on some PCs.)

c. Enter the following command, substituting the name of your server:

```
C:\>ping nrcs-a
```

A message similar to the following should appear:

```
Pinging nrcs-a [10.1.38.30] with 32 bytes of data:

Reply from 10.1.38.30: bytes=32 time<10ms TTL=255
Reply from 10.1.38.30: bytes=32 time<10ms TTL=255
Reply from 10.1.38.30: bytes=32 time<10ms TTL=255

C:\>;
```

If you do not get replies, then the server is either down or a networking problem exists. A message similar to one of the following may appear:

```
Bad IP address nrcs-a
```

-OR-

```
Bad host nrcs-a
```

The iNEWS Workstation is unable to look up the name to obtain the IP address of the server. It is unable to resolve the host name. This indicates the name entered in the login dialog box does not exist in the local hosts file on the PC or on the Domain Name Server (DNS) if your network is configured to use a DNS.

The `workdebug` command may be used in system operator or superuser mode.

2. If the workstation can successfully ping the server, the next step is to run the `workdebug stat` command on each server at the console.

The `workserver` process runs on iNEWS Servers that manage the connection between PCs and servers. The `workdebug stat` command queries the `workserver` to obtain a list of how many sessions are configured and how many are currently logged in:

```
NRCS-A# workdebug stat
workserver: configured GUI general count:A 19 B 0 C 0 D 0
workserver: running GUI general count:   A 0  B 0 C 0 D 0
(etc)
```

If `workdebug stat` does not return a response at the console, this may indicate the `workserver` process has either died or entered a non-responsive state. The `workserver` process may need to be killed and restarted. Contact Avid Customer Support for further instructions.

3. If you get a response from `workdebug stat`, you can put `workserver` into a diagnostic mode where it will display messages on the screen when PC clients attempt to establish a connection. To do this, select *all* servers and type:

   ```
   NRCS-A$ workdebug debug
   ```

   The message returned when the workstation attempts to log in may give you a clue as to the problem. If no messages appear when the workstation tries to log in, then the workstation is not reaching the server. Use the `workdebug` command again to turn off debugging mode and stop the diagnostic messaging, by typing:

   ```
   NRCS-A$ workdebug silent
   ```

## A User Cannot Access an Item

If a user cannot read or write stories in part of the database, he or she may not have permission to do so. Check the user's group permissions.

### Group Permissions

**To assign permission to a user:**

1. Use the `list g <username>` command to determine which groups the user belongs to.
2. Compare assigned user groups to groups assigned to the directory or queue. To do this, use the `list d-g <name of queue>` command.
3. If necessary, add the user to the appropriate group story or stories in `SYSTEM.GROUPS`. See Chapter 6 in the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information.

If a user can create new stories in a queue, but not edit existing stories, it may be because existing stories were created when the queue had another write group assigned to it. Previous stories would have been created with previous security restrictions. To change the group permissions on existing stories, use the `gtraits` command in the following format:

```
gtraits c <name of queue>
```

A user may also be unable to access a queue or story in the database if another user is ordering that queue or editing the story. The queue or story becomes “free” when the other user finishes the operation, but until then, it is considered to be “busy.”

## Busy Stories

Only one user can edit a story at a time. No one else can edit that story until the first user is done with it. If a second user tries to edit a story that another user is working on, the second user will get a message that the story is busy. When a user opens a story for editing, the system puts an edit lock on the story and removes it when the user saves the story and gets out of it. Edit locks prevent multiple users from making changes to stories.

Similarly, when a user goes into order mode, the system puts an order lock on the queue. No other user can change the order of stories in the queue until the first user exits order mode.

If a user is editing a story and the system crashes, or the user's PC locks and needs to be rebooted, the edit lock placed on the story remains attached to the story. When the system comes back up or after the user logs back in, he or she may be unable to edit the story. The edit lock will prevent anyone from making changes to the story and users trying to open the story will get a `story busy` message.

The story must be unbusied before any user can get back into it.

This is also true of order lock. If a user's PC crashes while he or she is in order mode, the order lock remains behind. Both edit locks and order locks are removed with the `unbusy` command. The syntax is:

```
unbusy <queuename>
```

You must know the exact queue name to unbusy it, such as `SHOW.6PM.RUNDOWN`.

When you unbusy a queue and there is an order lock on it, you are first asked whether you want to remove the order lock. Then you are asked whether you want to unbusy each busy story in the queue.

You can respond with `yes` to remove the edit lock or order lock, `no` to skip that story, or `quit` to exit.

**Care should be taken when removing edit locks. Do not unbusy stories that users are still working in. If you do, when they try to save the story it will be saved to the Dead queue.**

You may see messages on the console about edit-locked (busy) stories in the Dead queue. A large number of edit-locked stories at the bottom of the Dead queue can cause problems, and they should be unbusied.

The Dead queue is usually large and an unbusy on the entire queue could take a long time to execute.

### To unbusy edit-locked stories in the Dead queue:

1. Enter superuser mode on the console.
2. Use the `unbusy` command to remove the edit locks.
   ```
   NRCS-A# unbusy -i dead
   ```

The `-i` option allows the `unbusy` command to ignore the inverted attribute.

3. The system will prompt you for confirmation. After answering “yes” to unbusy the stories, watch for when the system is no longer finding busy stories. You can then break out of the unbusy process with **`Ctrl+\`** (Control-backslash) or use the Delete key to “interrupt” the unbusy process.

# Wire Problems

This section explains common problems users may encounter with their wire ingests during daily operation.

## Data Receiver

The Avid Data Receiver provides an Administration application that may be used to troubleshoot most issues related to ingesting wires to iNEWS using a Data Receiver instance. For more information, see the *Avid Data Receiver Installation and Operations Manual.*

If stories are not being sent to iNEWS—that is, the number of stories processed never increments in the Administation application—ensure raw data logging is enable and functioning. If it’s not, check your hardware connection and the Data Receiver configuration. If some raw data is being logged, but stories are still not being sent to iNEWS, then check the start/stop sequence configuration for Serial and Telnet input types. Also ensure file logging is set to ALL and check the log file for parsing errors. This would indicate a possible problem with parsing NSML, which could make iNEWS reject stories when Data Receiver attempts to send them.

## PCU Wire Ports

Wire ports are either on or off. They either receive data and stories or they do not. The wire port cannot decide to let some stories in but not other stories. If stories are coming in to your AP wire port but it is missing

some—weather stories for instance—this is most likely a problem with what AP equipment is feeding because the PCU port is not selective about which stories it decides to keep and which to ignore.

If a wire stops flowing into the system entirely, the first step is to figure out which wire port is not working. On its line in the `/site/config` file, each wire port is assigned a name, such as AP, RT, or NC. This name is in the fifth column of the wire's configuration line. For instance, the following configuration file entry shows a wire port with the name AX:

```
wire    23    9600-8n   anpa7  AX - ;Associated Press
```

You can check `WIRES.ALL` and look at the coding on the stories to determine which port is not flowing, and then restart that wire port. For the example above, stories that come in port 23 will have coding that begins with AX in the writer field:

```
Title                        Writer       Created
AP-IN--FWA-SevereThunder   AXv2in-gob   07/06/00 14:54:53
```

If you notice AX-named wires had not come in for a period of time, wire port 23 must be restarted and then the Wires queues monitored to see whether the wire is working again.

Alternatively, you can restart the wire port with the minus four (`-4`) option:

```
NRCS-A$ restart -4 23
```

When restarted in this diagnostic mode, a message is printed to the console screen each time a wire story arrives and is placed in a Wires queue. Restart it normally once you have confirmed wire flow.

If it is still not flowing, get instruction from your wire service provider on how to reset their equipment. Resetting the wire's modem should trigger a wire reset message which should arrive in either the `WIRES.ALL` or `WIRES.UNKNOWN` queue. If the reset message reaches the database, it is an indication the wire port is working, and wire stories should then start to flow.

If you do not get wire flow, try stopping, resetting (power cycling), and then restarting the entire PCU.

You can also do a wire dump, which redirects any characters coming in the PCU port and displays them on the console screen. If you are getting no wires and no wire dump, it may be possible the wire modem is feeding the signal with the wrong polarity.

The only way to determine polarity is to hook up a mini-tracker and examine the pin 3 light as data is received. Pin 3 should flicker green as data flows across the connection. The PCU port will not recognize red data flickering. Red data indicates incorrect polarity coming from the wire equipment and must be rectified by your wire service provider.

*Red data will appear on terminals and printers hooked to the wire output, but it will not be recognized as data by the PCU port.*

Each wire port must have a corresponding wire profile in `/site/wires/<port #>` if you need to move a wire to another PCU port. For instance, if you move a wire from port 17 to port 38, you must also copy its profile to the new port number with the Move (`mv`) command:

```
mv /site/wires/17 /site/wires/38
```

# System Printer Problems

When a story is printed to a system printer, the print job goes to the appropriate `SYSTEM.PRINTERS` queue.

The printer port communicates with the printer using xon/xoff flow control. The printer port starts sending a print job out the port. If the printer's buffer is getting full, the printer sends an xoff signal to the port, telling it to stop sending data. When the printer is ready to accept more of the print job it sends an xon signal, telling the port to resume data flow.

If the port does not receive an xoff signal, it assumes the printer is processing the print job and pipes it out the port. It stops only if the printer tells it to stop.

If the printer is off-line, it will transmit an xoff signal back to the port when the port tries to send a print job, and print jobs will back up in the `SYSTEM.PRINTERS` queue. The system will try to print the job for up to

three minutes. After that time a message will be returned to the user stating the printer is off-line and needs attention. A similar message appears on the console.

Occasionally, a user will print an entire queue, such as `WIRES.ALL`. This is called a *runaway* print job and it must be deleted from the `SYSTEM.PRINTERS` queue.

**To eliminate a runaway print job:**

1. Power off the printer or put it offline so it stops wasting paper.
2. Kill the print jobs in the `SYSTEM.PRINTERS` queue.
3. Restart the printer port.
4. Turn the printer back on.

*You must restart the printer port because the print job has already been downloaded to the PCU for printing. If you kill the print jobs and then turn on the printer, it will resume printing the job.*

Each printer port must have a corresponding printer profile in `/site/printers/<port #>`. For instance, if you move a printer from port 26 to port 41, you must also move its profile to the new port number:
`NRCS-A$` **`mv /site/printers/26 /site/printers/41`**

# Locked Blocks

During normal operations the system constantly locks and unlocks blocks on the hard drive as data is accessed. Occasionally, something may go wrong and a locked block is left behind.

If you have a persistently locked block, any other process that tries to access the locked block will wait for the block to become available. The usual scenario for this problem is that PCs start hanging in the newsroom when users get to a story in a rundown that is in a locked block. Users then move to another PC, and hang that one too when they access the locked block story.

If the original locked block is unlocked, then all frozen PCs “unhang.”

To check for a locked block, type the `dblock` command on all servers. Type `dblock` several times in succession and note the block numbers reported. If block numbers are changing, then the process is not hung up on a locked block. If the block number and process ID number remain the same through successive `dblock` commands, then it may be a locked block situation. If a block remains locked for more than one minute, the system sill print a diagnostic on the console and will repeat the message every minute the block remains locked. The system will remove block locks owned by processes that have exited.

If you have a persistently locked block, contact Avid Customer Support.

# How to Check Process Status (ps Command)

Occasionally, the support staff may ask you to run a few Linux commands to troubleshoot problems. One of the most common would be a request to check on the status of a process using the `ps` command.

```
NRCS-A$ ps -ef
```

The `ps -ef` command returns a list of all the processes currently running on the server. On large systems there can be hundreds or even thousands of processes.

The process list gives useful information about each process including the process owner, when it was started, on which terminal port it started, how much processor time has been expended on the process, and so forth.

You can also use piping with the `fgrep` command to search the process list for a particular process and check whether it is running. This example produces a full process list and then filters it for the lines that contain the word, workserver:

```
NRCS-A$ ps -ef | fgrep workserver
```

When you use `fgrep` on the process list for a specific word, two processes are usually returned in the display. One line is the process you are looking for, and one line is the `fgrep` process that looked for that word.

If you know a particular process ID number, you can check to see if that process is running or has completed:

```
ps -p <pid #>
```

If the process is still running, you will get a one-line process list for that number:

```
PID TTY      TIME CMD
516 tablet   0:01 workserver
```

If it is not running, the command will simply return a header line like this:

```
PID TTY      TIME CMD
```

You may also use `ps -fp` for a fuller listing.

# Power Failure

If you experience a power failure, the servers will reboot. After they work their way back to the login prompt and you have logged in, the servers will not be named, they will be at the question mark-colon prompt.

```
?:
```

If both servers went down at the same time, the databases will still be in mirror and they can be connected normally and started up:

1. On *all* servers simultaneously, type:
   **`connect #`**
2. To remove edit locks, on the master computer only, type:
   **`dbclean -x .`**

*This command is not strictly necessary. "Stale" edit locks are detected and ignored. An edit lock is considered to be "stale" if the session that owns the lock no longer exists or if the session was started—logged in—after the edit lock was acquired.*

3. On *all* servers simultaneously, type:
   **`startup`**

If they are plugged into different UPSs and they ran through the UPS battery and then lost power and rebooted, you will not know if they went down concurrently. One UPS may have run longer than another. If so, more stories may have flowed in on a wire or been saved by a user while the one server was still up and the other was down. The database will not be in mirror and you will need to go through the recovery process as noted in Chapter 1.

# Network Failure

The iNEWS newsroom computer system is a networked client-server application. A well-running Ethernet network is essential for proper communication between the devices. A network failure can disable the entire system.

The simplest way to test network connectivity is to try to `ping` other computers on the network. When you `ping` a server, your computer sends a “pulse” across the network. The pulse then is echoed back from the target server and returns to the sending computer in the form of ICMP replies. If the network or server is down, you will not get replies from the server (or computer) you are trying to ping.

```
NRCS-A$ ping 152.165.17.110
```

-OR-

```
NRCS-A$ ping nrcs-a
```

In the previous ping command examples, the number of ping requests sent would continue endlessly. To control the number of ping requests use a specific number to limit the number of requests. For instance, in the following examples, five ping requests will be attempted as noted by including the number **5** in the ping command.

```
NRCS-A$ ping 5 152.165.17.110
```

-OR-

```
NRCS-A$ ping 5 nrcs-a
```

In the case of an iNEWS Workstation that is having problems connecting, the first step is to try pinging the server from the workstation to make sure they can see each other on the network. (The syntax for `ping` on Windows is different than the Linux `ping` command.)

If computers are unable to ping each other, check for a loose or disconnected network cable or a hub that may have lost power.

## netstat -i Command

One diagnostic command you can run on the server to quantify network errors is the `netstat -i` command. The `netstat -i` will show:

- How many packets have been transmitted
- How many input errors (`Ierrs`) have been detected
- How many output errors (`Oerrs`) have occurred

## Output Errors (Oerrs)

The primary barometer of network health is the Output Errors (`Oerrs`) column. The computer will try to transmit a packet 15 times before chalking the attempt up as an `Oerr`. It will then try to transmit again 15 times and may wind up incrementing the `Oerr` count. Clean networks will show zero `Oerrs`, or no more than a few.

Of more concern than the raw number of `Oerrs` is how quickly they are increasing. If you are picking up an `Oerr` every minute, this would be indicative of network problems and a disconnect may be imminent.

Periodically run a `netstat -i` so you can get a baseline feel for how many `Oerrs` your system produces each week or month.

## Input Errors (Ierrs)

Input Errors (`Ierrs`) are fragments of packets or unrecognizable packets. Systems co-existing on a Novell Local Area Network (LAN) generally show many `Ierrs`, although they do not seem to cause problems.

# iNEWS and FTS Connectivity (ftsdebug Command)

For more information about Fast Text Search and its utility programs, see Chapter 13 in the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

The ftsdebug command can be used to verify communication between the iNEWS Servers and the two server (utility) programs related to Fast Text Search (FTS). It can also be used to compare the contents of indexes and indexed queues, adjust the indexes, and is a vital tool for FTS troubleshooting and operation. The format for the command is:

```
ftsdebug [index | search] <parameter>
```

The `index` option will apply the `ftsdebug` command to the utility program used by FTS, known as the ftsindex server or `ftsidx.exe`. The `search` option will apply the `ftsdebug` command to the utility program known as the ftsseek server or `ftssch.exe`.

| Parameter | Description |
|---|---|
| closelog | Close the log file (leaves log intact). |
| debug | Set debug log level for the program to:<br><br>Level 0 - no logging done<br>Level 1 - each action logged<br>Level 2 - request parameters logged<br>Level 3 - request parameters and log each action |
| openlog | Open/reopen the log (truncates existing log). |
| shutdown | Shutdown the program on the FTS server. |

*There is no start parameter because the FTS programs must be started from the FTS server.*

| Parameter | Description |
|---|---|
| statistics | Show program version, start time, and statistics. |
| viewlog | Print the contents of the log file |
| **Limited to index only:** | |
| showrequests | List pending index requests (`SYSTEM.INDEX`) |
| tune on \| off | Turn tuning on or off or force a major tune. |
| tune <type> | Types of tunes are: minor, major, and absolute (which tunes immediately) |
| **Limited to search only:** | |
| adjust [-v] <path> | Compare FTS index for <path> against database content and fix discrepancies. Used to bring indexing up-to-date. For instance: **`ftsdebug search adjust archives.2002`** |
| compare [-v] <path> | Compare FTS index for <path> against database content. |
| list [-n] <path> | List FTS index entries for <path> |

## Checking Communication between iNEWS and FTS Servers

Verify communication from the iNEWS Servers to the FTS utility programs running on the Windows-based FTS server by using the following forms of the `ftsdebug` command:

**`ftsdebug index stat`**

**`ftsdebug search stat`**

These commands may also be used to monitor the indexing process.

After each command is entered at the console, a display of statistics about the utility program, either the ftsindex server (`ftsidx.exe`) or the ftsseek server (`ftssch.exe`), appears. The display will vary depending on which form of the command you used. Samples of each are shown in the following two columns:

```
NRCS-A$ ftsdebug index stat
Program name: ftsidx.exe
Version: 1.5.0.14 FTS
Process id: 238
Started: 07/10/04 09:00:52
Debug value: 0
Tuning: on
Index Base: e:\fts\nrcs-a
Indexed Documents: 1458
Requests Serviced: 7365
Documents Added: 1458
Documents Removed 81301
Documents Replaced: 2
Documents Purged: 1
```

```
NRCS-A$ftsdebug search stat
Program name: ftssch.exe
Version: 1.5.0.14 FTS
Process id: 239
Started: 07/10/04 09:00:52
Debug value: 0
Index Base: e:\fts\nrcs-a
Indexed Documents: 1458
Requests Serviced: 39
```

The number of requests serviced includes index requests, purge requests, and ftsdebug status requests, so it is unlikely to match the number of indexed documents.

If communication with the FTS utility programs on the FTS server is unavailable, the `ftsdebug` command will hang, and then eventually time out and respond with a message that it could connect. If this happens, check the Windows-based FTS server to see whether it rebooted. Also, confirm whether the utility programs (`ftsidx.exe` and `ftssch.exe`) are running. Stop and restart them, then check for communication again.

*If the* `ftsdebug` *command hangs, you can break out of it by pressing the Control and C keys (Ctrl+C).*

# Appendix A

# Command References

Most of your system's commands are special commands provided by Avid. The commands you are most likely to need are listed and explained in this appendix, along with examples.

**Some available commands are meant to be used only by Avid technicians or under the supervision of Avid personnel. These commands may cause damage if used improperly. They are listed in this appendix on page 114.**

This appendix contains the following main sections:

- Programs Invoked by iNEWS
- Commands Used by Avid Personnel Only
- Linux Commands Used in iNEWS
- Console Control Commands
- Console Server Commands
- Job List Commands
- Dialog Commands

# Programs Invoked by iNEWS

The following programs are invoked and used by your iNEWS system. Do not use them as commands.

| | | |
|---|---|---|
| action | ftsseek | nxserver |
| bio.conf | gnews | parallel |
| bioserver | ismessage | prtscanner |
| boot | keyword | rxnet |
| brand | license | seek |
| cgi-fcgi | mailserver | server |
| connect.sh | mct | snews |
| console | monitor | start |
| copyright | monserver | telex |
| disconnect | news | txnet |
| distribution | newsmail | webaccserver |
| ftsindex | NewsWeb.fcgi | workserver |

# Commands Used by Avid Personnel Only

The following commands are used by Avid personnel only.

| | | |
|---|---|---|
| attach | dblock | msgdebug |
| bdump | dbprint | nsupgrade |
| binhex | dbsize | poke |
| biodebug | detach | qcheck |

| | | |
|---|---|---|
| biosleep | diddle | qstampcheck |
| biostat | download | qxcheck |
| catcheck | finit | rcat |
| ccuputwire | ftsdebug | traverse |
| ccuq | ifis | unjumble |
| ccureset | ifmaster | userclean |
| ccuspeed | ifsu | wdump |
| ccutime | iftapeis | workdebug |
| dbcopy | keycheck | wxlate |
| dbgen | kwdcheck | xi |
| dblinks | load | |

# Linux Commands Used in iNEWS

The following Linux commands are available in iNEWS. For further information, see the reference material that came with your Linux system. To obtain command syntax and other usage information, type the `man` command along with the command name. For instance, to get information about the `grep` command, type:

```
man grep
```

| | | |
|---|---|---|
| cat | kill | pwd |
| cp | more | rm |
| date | mv | sync |
| df | passwd | telnet |
| grep | ps | |

# Console Control Commands

The table below lists available console control commands and their functions.

| Keyword | Explanation |
|---|---|
| `bottom` | Moves you to the newest (bottom) line in the history. |
| `computer [`*`computer-name(s)`*` \| *]` | Selects one or more of your system's servers, so that you can type a command on the selected server(s). The command may be abbreviated: typing **`c`** or **`computer`** plus the computer name(s) of the servers you want to select will work. Type the command plus an asterisk (*) to select all servers, such as **`c *`**. |
| `down [`*`number-of-lines`*` \| `*`keyword`*`>` | Moves you that many lines forward (down) in the history.<br>• `down` *`number-of-lines`* moves you that many lines forward in the history. For example, typing **`down 30`** moves you forward 30 lines. If you follow `down` with a number greater than the number of lines between your current position and the last line in the history, the `down` command moves you to the last (newest) line in the history.<br>• `down` *`keyword`* searches forward through the history from your current position for that word. For example, to search forward for a line containing the word, `list`, type: **`down list`**.<br>The `down` command is not case sensitive. If you specify `PEOPLE`, `people` is a valid match. If the system does not find the keyword before reaching the bottom of the history, the console beeps and you are returned to the current line.<br>• `down` with no parameter moves you down one line.<br>A wildcard character (#) can be used to match any character or to a search for a number. For instance, `down 160#` will search for the number `160` instead of moving down 160 lines. |

| Keyword | Explanation |
|---|---|
| `Exit Function` | **To leave the console program and return to the MS-DOS prompt:**<br>1. Press the Command key.<br>2. Press Ctrl+E.<br>3. Press Enter.<br>(You cannot type `exit` at the Command prompt.)<br>Type **`console`** at the prompt to restart the console program. |
| *`function-key-number=definition`* | Assigns a command to a console function key. For instance, to assign the `list s` command to the F9 key, type **`f9=list s`**. To include Command and Enter keys in a function key definition, use braces. For instance, to include those keystrokes in the previous assignment example, type **`f9={list s}`**. To list the current assignment of a function key, type the key number by itself plus Enter on the command line. The valid range is `f1-f14`. Avid recommends no definition be made for F14, which defaults to the Command key. F13 corresponds to the plus (+) key on the numeric keypad. |
| `list [`*`#-of-lines`*` \| all]` | Sends some number of lines of the history to the printer attached to the console multiplexor.<br>• `list` followed by a number, such as `list 3`, prints that many lines of the history beginning at the current line.<br>• `list all` prints everything from the current line to the newest line.<br>• `list` with no parameter prints the current line.<br>When using `list`, the word `PRINTER` appears on the command line. If `PRINTER` is displayed but nothing is being printed, ensure that the printer is plugged in, turned on, online, and has paper. The console assumes a printer is connected to the PC's parallel port, where output is sent. To cancel a `list` command while output is printing, press any key; printing stops and your position in the history moves to the last line sent to the printer. |
| `logclose` | Writes all history currently in memory to disk and then disables disk logging. You can use it to change log disks. |
| `logopen` | Resumes history disk logging after it has been disabled with `logclose`. |

| Keyword | Explanation |
|---|---|
| `logout` | Logs you out from a remote console. To log out, press the Escape key and type **`logout`**. The remote console displays the `PASSWORD` prompt, and you can then hang up your modem. |
| `modem` | Typed at the main console, this command logs out a remote console user. Before you type this command, ensure that the remote console user is not in the middle of an operation. The command may be abbreviated: typing **`m`** or **`modem`** will work. |
| `print [`*`number-of-lines`*` \| all]` | Displays a number of lines of the history on the console screen.<br>• `print` followed by a number displays that many lines of the history beginning at the current line.<br>• `print all` displays everything from the current line to the newest line.<br>• `print` with no parameter displays the current line.<br>To cancel a `print` command while it is displaying console history, press any key. The console stops at the line last displayed on the screen, and your position in the history moves to that line. |
| `reset` | Attempts to unfreeze one or more of the console's I/O ports, if communication has failed between the console and your system's servers.<br>To use `reset`, select the servers that are affected, press the Command key, and type the command. The command may be abbreviated: typing **`r`** or **`reset`** will work. |
| `top` | Moves you to the oldest line in recent console history. To see history older than this, use the `view` command to view a history disk log file. |

| Keyword | Explanation |
|---|---|
| up [*number-of-lines* \| *keyword*] | Moves you backward (up) some number of lines in the history.<br>• up *number-of-lines* moves you that many lines back in the history. For example, typing **up 30** moves you back 30 lines.<br>If you follow up with a number greater than the number of lines between your current position and the first line in the history, the up command moves you to the first line in the history.<br>• up *keyword* searches backward through the history from your current position for that word. For example, to search backward for a line containing dbpurge, type **up dbpurge**.<br>The up command is not case sensitive. If you specify PEOPLE, people is a valid match. If the system does not find the keyword before reaching the top of the history, the console beeps and you are returned to the current line.<br>• up with no parameter moves you back one line.<br>A wildcard character (#) can be used to match any character or to a search for a number. For instance, up 160# will search backwards for the number 160 instead of moving up 160 lines. |
| view <*drive:filename*> | Displays a DOS text file on your console screen.<br>Use this command to look at old history that has been saved to disk or to read other disk files, such as the console configuration file. Before you type this command, use the zoom command to display only one server's region on the console screen. (You can choose any server.)<br>While you are viewing a file, you can only move down through it (as opposed to back or up in the file). Each time you press the Enter key, the file scrolls down one line.<br>To stop viewing the file and return to normal console operation, press the Command key.<br>The command may be abbreviated: typing **v** or **view** plus the filename will work. |
| x | Restarts the sending and receiving of information by the computers' console ports.It stands for XON and causes an XON character to be sent to the server for each selected region. If you are having trouble communicating with your servers from the console, try this command. If it does not work, use the reset command. |

| Keyword | Explanation |
|---|---|
| `zoom <computer-name>` | Selects a single server and fills the console screen with its region. To return the screen to its normal split-screen, multi-region state, select any server with the `computer` command. The command may be abbreviated: typing **`z`** or **`zoom`** plus the computer name will work. |

*For commands that include an individual server letter (A, B, C, or D), use the pound (#) character to send to multiple servers without typing them individually; The console will convert the # to the server's letter, such as A or B. For instance, if you type* **`connect #`***, the console sends the command* `connect A` *to server A,* `connect B` *to server B, and so forth. To enter a # without it converting to a server letter, press the Enter key on the numeric keypad, then press Shift+3.*

# Console Server Commands

You must enter commands in lowercase. Your system does not recognize commands entered in uppercase.

`broadcast` `broadcast <message>`

Sends a message to everyone logged in. The `-d` option will issue the broadcast in a popup window. For instance, to send a message, select *one* server and type:

`NRCS-A$` **`broadcast -d System going down at 12:00`**

*Certain characters are interpretted by the bash shell program, so when including characters, such as angled brackets, exclamation points, asterisks, or pound signs (`<`,`>`,`!`,`*`,`#`) in the* `<message>`*, the entire message should be placed within quotation marks to prevent the program from interpretting them.*

`configure` `configure [-ns] [<config file> [<system> <computer>]]`

Incorporates changes to your configuration file into your system's operation, and checks the configuration file for any errors.

For instance, suppose you made changes to PCUs 10 and 20, which are connected to server A in an AB system. To test these changes, become a superuser and type:

```
NRCS-A# configure /site/config ab a
```

If no <config file> is specified, /site/config is used.

If a service has been added to a database story in SYSTEM.SERVICE, use configure -s so the service can be recognized.

If an Ethernet or Internet address has been added to a database story in the SYSTEM.CLIENT.WINDOWS directory, use configure -n to validate the address and allow it to be recognized by the system.

In both cases, you must first take the system offline, enter the configure command, then put the system back online.

### connect

```
connect  <name>  [<option>=<value>] …
```

The connect command names each server in the system, and tells each how many other servers there are in the system and how to communicate with them. For instance:

```
NRCS-A$ connect a net=ab
```

*The* connect *command reads the* /site/system *file for options. If an option is specified on the command line and also in the* /site/system *file, the command line setting takes precedence.*

**connect Command Options**

| | |
|---|---|
| auto_upgrade=<yes \| no> | msgserver=<silent \| verbose> |
| clockmax=(12 \| 24) | name=<a \| b \| c \|d> |
| disk=<status> | pausetimeout=<mm:ss> |
| excludedvideo=(director \| none) | purgelimit=<number of hours> |
| highwater=<number_of_blocks> | readrate=<number of words per minute> |
| id=<system name> | remotetimeout=<mm:ss> |

| | |
|---|---|
| lastlogin=<yes \| no> | scriptlhmax=<number of columns> |
| load=<number> | scriptrhmax=<number of columns> |
| localtimeout=<mm:ss> | security=<and \| or> |
| lowwater=<number_of_blocks> | single=<name> or net=<name,name[,name,name]> |
| master=<a\| b \| c \|d> | timechar=<character> |
| maxhits=<number> | timer=<silent \| verbose> |
| min_passwd_length=<number> | wordlength=<number> |

`dbclean`

`dbclean [-x] <directory name>`

When starting up after a power failure, use `dbclean` to remove any edit or order locks in the database. Run the command before startup or log everyone off the system by typing **`logout all`** before issuing this command.

The most common usage of `dbclean` scans all queues except those marked with the skip flag. To use this command, after logging out all the users, type:

```
NRCS-A$ logout all
NRCS-A$ dbclean -x .
```

`dbclose`

`dbclose`

Closes the database.

**If you use this command while users are active, changes to stories will be lost.**

`dbdev`

`dbdev`

Reports the disk partitions in use and the number of blocks allocated for the database on each disk partition. To find out the size of your database, type:

**`dbdev`**

A message similar to the following appears:

```
/dev/rp5  0        60000
/dev/rp6  60001    120123
/dev/rp7  120124   999999
```

The numbers reported are the first and last blocks for each partition. The last number of the report (999,999 in the example above) is the size of the database.

`dbdump`

`dbdump <keys> [<option>] …`

Dumps individual stories or the entire database to tape.

This command can be interrupted. The program will continue dumping to reach an appropriate quitting point.

| Valid Keys | Description |
|---|---|
| a | Append to current dump |
| c | Create new dump |
| C | Create new dump, don’t ask if it’s ok. |
| d | Dump the news directory |
| i | Dump indexed files (such as, user index) |
| s | Show quick index of dumped stories |
| v | Verbose output |
| x | Ask before dumping indexed file |

| Valid Options | | Description |
|---|---|---|
| -a | <device> | Use alternate device for dump |
| -d | | Dump news directory skeleton (no stories) |
| -f | [file] | Dump to file (use '+' for standard output) |
| -m | <minutes> | Dump files modified in last *x* minutes |
| -n | <namelist> | Only dump listed directories |
| -N | <computer name> | Network dump to specified computer |

To dump everything except those directories marked with a skip flag, type:

**dbdump c**

To dump a queue to a new tape, add `-n` and the queue name to the command. For instance, to dump the queue `SCRIPTS.JUNE.01` to a new tape, type:

**dbdump c -n scripts.june.01**

By replacing the `c` with an `a`, you can add a queue to a tape without erasing information already on the tape. For instance, to append `SCRIPTS.JUNE.10` to a tape, type:

**dbdump a -n scripts.june.10**

Dumps already on the tape are skipped and this dump is added to the end.

## dbfree

`dbfree [-cf]`

Reports the size of the database and the size of the free list—that is, the amount of free blocks available. To display this information, type:

**dbfree**

A message similar to the following appears:

```
data base size 4106240, free 72600=18%, freelist 2904
```

*The freelist is compressed, so there can be as many as 5000 free blocks accounted for in each freelist block.*

To display the amount of free space in your software area, type:
**df**

Prior to the system being configured, there are two maintenance options that may used with the `dbfree` command.

The format is `dbfree [-cf]`.

The `c` option is used to check for cleared free blocks. The `f` option is used to "unclear" cleared free blocks.

`dblines` `dblines [b | c | f | n | q | s | v |+S|-S|-O] <pathname>`

Checks the database for story errors.

| Valid Keys | Description |
|---|---|
| b | Story block count only (no checking) |
| c | Complete check |
| f | Compare fields in story form with fields in story. Report those not found in both. |
| n | Do *not* fix errors |
| -O | Do *not* report orphaned story record errors |
| q | List queue names |
| s | Skip queues that are skipped by `dbdump` |
| +S | Fix script related errors |
| -S | Skip (not report) script related errors |
| v | Verbose output |
| w | Check word counts |

| Valid Keys | Description |
|---|---|
| x | For qstamp checking |

*The dblines program normally will not fix script related errors; if you want to do so, the* `+S` *option must be specified and the* `n` *option must not be specified.*

Use this command weekly as part of normal database maintenance. Start `dblines` before you go home and run it in the background. To run `dblines` in the background, precede `dblines` with a `print` command. Including a period with the command checks the entire database.

For instance, to run `dblines` in the background and send any error messages to system printer 2, type:

**`print 2 dblines`**

If `dblines` finds any errors related to queue corruption, call Avid Customer Support for assistance.

`dboriginal`

`dboriginal [-a] <pathname>`

Removes all old versions of stories in a queue to the freelist, so use it only on queues where you do not need to retain these old versions. For instance, to remove the old story versions in `ARCHIVE.MARCH`, type:

**`dboriginal archive.march`**

Use the `dboriginal` command to reclaim space when the system is low on space.

The `dboriginal` command will print diagnostics, indicating how many stories were examined and how many old versions were removed. The command removes old versions for unshared stories—those with a link count of 1—in a queue. The option, `-a`, allows the command to remove old versions from all stories (shared and unshared).

*The* `dboriginal` *command can also be used to remove old versions from stories in a queue that has the* `save none` *attribute. When stories are moved to or copied to a queue, all old versions are retained even if the queue has this attribute; only when a story is edited are the old versions removed.*

As indicated by the pound sign (#) in the prompt, you must be in superuser mode to execute commands. Also, you must use 24-hour military time.

It should only be necessary to run this command once or twice a year. Below is an example of using the `at` command to execute a `dboriginal` command at a specified schedule and directing it to a specified area (`archive`).

```
NRCS-A# at 11:10
/exc/dboriginal archive
<Ctrl+D>
job 1001340809.a at Sat Jul 10 11:10:00 2004
NRCS-A#
```

`dbpurge`
(Superuser conditional)

`dbpurge <path> [- h | l | f] [<interval>]`

`dbpurge` purges the database to regain space.

| Valid Keys | Description |
|---|---|
| - | Condition used as `<path>` parameter. Purges all queues with the default interval |
| -v | Condition used as `<path>` parameter. Purge all queues listed in the command in verbose mode |
| -h | Include held entries (must be superuser to use this option) in the purge |
| -l | Include locked entries (must be superuser to use this option) in the purge |
| -f | Include future dated entries (must be superuser to use this option) in the purge |
| | Interval is expressed in `<hours>` or `<days>.<hours>` |

*Any queues that have a write group assigned will not be purged when* `dbpurge` *runs as a non-superuser.*

The system automatically purges the directories and queues each hour to move old material from high-turnover queues (such as the Wires queues) into the Dead queue. Normally locked and held stories are not purged regardless of their age. Use this command only in an emergency when you need to regain some space. Stories can receive a date in the future if your system date is inaccurate. If you have future-dated stories because the system date was inaccurately set, remove them using the `-f` option or wait until the date expires. You must be a console superuser to run `dbpurge` on write-protected queues. For instance, to purge all stories in the Wires queues older than five hours, type:

```
dbpurge wires 5
```

Typing `dbpurge -` purges all queues in the database according to each queue's purge interval. `dbpurge -v` does the same, and prints a message on the console for each queue purged.

You can also use `dbpurge` to remove held and locked stories from the database. To remove all locked stories from all queues in the People directory, type:

```
dbpurge people -l
```

To remove all held stories from a queue, use `dbpurge`, but substitute `-h` for `-l`.

*The* `dbpurge` *command can be run only on the master computer.*

`dbrestore`

```
dbrestore <key> [option] ...
```

The `dbrestore` command restores data dumped via `dbdump` to the iNEWS database. Automatic data conversions are performed to convert data from older database revisions to the current database format.

Conversion from the Avid NetStation database format is controlled with conversion parameters stored in a database conversion profile file —normally, `/site/dict/conversion` or optionally specified with the `dbrestore -c` option). For NetStation dbdumps, existing form

templates stored in directory records and in stories will be converted to form names consisting of three digits. See the `qform`, and `sform` database traits.

If your database is damaged, you can restore it from your backup tape by typing:

```
dbrestore div
```

When you use `dbrestore`, restoring large numbers of stories can cause a temporary out-of-space condition.

Press Delete to stop a `dbrestore` in progress.

### dbrestore Command Key Options

| | |
|---|---|
| **Keys:** | |
| s | Restore stories only |
| d | Restore stories with their original directories |
| i | Restore ISAM files |
| v | Verbose output; list directory names |
| vv | List directory names and *each* title |
| x | Ask before restoring each ISAM file |
| f | Print facts about blocks and times |
| t | Print table of contents; do not restore |
| **Options:** | |
| -a <device> | Restore from alternate device |
| -c <filename> | Database conversion profile filename |
| -C <filename> | Character map filename |
| -d | <date>[-<date>] |
| -f [file] | Restore from a file (use '+' for standard input) |

| | |
|---|---|
| -i | Do *not* index. Do *not* post index request to `SYSTEM.INDEX` when a story is restored in an indexed queue. |
| -k | <keyword> (You can specify multiple keywords.) |
| -m <value> | Maximum number of stories to restore |
| -M | Preserve modification times |
| -n [=] | <directory>[=<new name>] |
| -N | Read from network socket for dump data |
| -p <queue> | Only with key letter s, will create queue |
| -s <platform> | Linux |

*You cannot use both the* `s` *and* `d` *options in the same command. Select one or the other; not both. Additionally, you must be a superuser to list ISAM files.*

## dbserver

`dbserver <high water>`

Reclaims space from the Dead queue and places it on the free list. Use `dbserver` to build up the free list prior to periods of peak use.

When you use `dbserver`, specify the total number of free blocks you want to have in the free list. If the free list contains 25,000 blocks and you want to build up the free list to 100,000 blocks, type:

```
dbserver 100000
```

*When the free list contains the desired number of blocks* `dbserver` *stops. If you specify a number that is smaller than the current free list size,* `dbserver` *will not do anything.*

You can also use `dbserver` to place all space in the Dead queue on the free list. To do so, specify that you want to rebuild the free list to an unreasonably large size.

For instance, on a maximum 16 gigabyte database, type:

```
dbserver 16000000
```

The `dbserver` command is invoked when the system is booted and runs in the background continually monitoring the number of free database blocks based on the high and low water marks specified in the system profile.

`dbsort`

```
dbsort  [-v]  <queue name>
```

The `dbsort` command sorts stories in any queue by the quick-index field, and should primarily be used to verify that the quick-index field accurately reflects the story sort field contents. For instance, if `RUNDOWN.AM` has the page-number field set as its quick-index field, type the following to sort the queue by page number:

```
dbsort rundown.am
```

If no quick-index field has been set for the queue, its stories are sorted by the value of the title field. If a sorted queue is ordered, the sorting is disabled. Using `dbsort` starts the sorting again. Only a superuser can sort queues with nonzero write groups.

*Dbtraits will automatically sort a queue when the sort attribute is turned on with the* `+so` *option. An ordered queue is automatically sorted when the ordered attribute is turned off with the* `-o` *option.*

Use the `-v` option to verify the sort field. The system checks that the quick-index field in the database has the same data as the sort field in the story. This option provides no sorting function, but it updates the quick-index field so that your next sort is based on current information.

`dbtraits`

```
dbtraits <pathname> [only] [<option> <value>]
[+|- mode] …:
```

**dbtraits Command Options**

| | |
|---|---|
| abstractlines or al | queueform or qform |
| abstractprinter or ap | readgroup or rg |
| abstractstyle or as | reindex |

| | |
|---|---|
| changeform or cform | save |
| displaylines | sortfield or sf |
| mailbox or mb | storyform or sform |
| notify or ng | stripform |
| purgeinterval <days.hours> | writegroup or wg |

Sets and modifies database traits.

To assign a story form called rundown to the `RUNDOWNS.5PM` directory, type:

```
dbtraits rundowns.5pm storyform rundown
```

To assign a queue form called rundown to the `RUNDOWNS.5PM` directory, type:

```
dbtraits rundowns.5pm queueform rundown
```

*Assigning forms can be combined to one command line, such as:*

```
dbtraits rundowns.5pm storyform rundown queueform rundown
```

**dbtraits Command Modes**

| | |
|---|---|
| f | Forms allowed |
| g | General |
| i | Inverted |
| index | Indexed |
| o | unorder (- only) |
| p | Printable |
| q | Queue operations allowed |
| r | Read access |
| refresh | Queue refresh |

| | |
|---|---|
| s | Sequential |
| so | Sorted |
| t | Text timing clocks |
| u | Update |
| w | Watch appends |
| x | Skip |

`dbvisit`

```
dbvisit  -<d | v | i> -[r | m name]  [-s]  [block# …]
```

**dbvisit Command Options**

| | |
|---|---|
| -d | Display progress by printing dots |
| -i | Validate ISAM file record and key sizes |
| -m | Machine name to disconnect (for online use) |
| -r | Read only; do not rebuild free list |
| -s | Operate in slow mode to eliminate cache usage |
| -v | Verbose output; print name of each queue |

*The validation is a validation of ISAM key and record sizes to prevent problems after software upgrades when the key and/or record sizes of any ISAM files changed for the new software release.*

Scans the database for errors, then rebuilds the free list and fixes bad story-link counts. A list of block numbers can be specified, as shown by the `[block# …]` parameter above. A diagnostic is printed whenever `dbvisit` encounters one of the specified blocks, which is helpful for tracking down database corruption. Use `dbvisit` once a month as a part of your regular maintenance.

There are two ways to run `dbvisit`:

- Without the `-m` option: The system must be offline and shut down. No one can log in until `dbvisit` is complete.
- With the `-m` option: Specify a machine that will be running `dbvisit`. The system must be offline and shutdown only long enough to enter the `dbvisit` command. Remaining machines can be brought up for users to log into. After the `dbvisit` procedure is complete, the machine running `dbvisit` can be reconnected following the normal procedure.

**Before you run `dbvisit`, ensure the system is offline, all PCUs are stopped, and the system is shut down. If `dbvisit` reports any errors, do not rebuild the free list; call Avid Customer Support for assistance.**

`dial`

```
dial -p<port> [-i<devid> -o<options> - b|c|d|e|q|S|t| -
m<modem> -n<phone#>]
```

This command makes a connection between a workstation and a modem, passing data in both directions. Use it only in the service table, not from the console.

**dial Command Options**

| | |
|---|---|
| -b | Binary data |
| -c | Check evidence of modem |
| -d | Direct (no modem) |
| -e | Echo |
| -i | Device ID for a direct-line device |
| -m | Modem type |
| -n | Phone number |
| -o | Communications options |
| -p | Port number |
| -q | Allow user to quit with Control <space> |
| -S | Special device |

| | |
|---|---|
| -t | Touch tone |

`dictionary`

`dictionary [-update] <dictionary> | <directory> ...`

Any combination of dictionary names and dictionary directories can be specified. For each directory specified, all dictionaries within that directory will be compared to their standard dictionary counterpart. Only missing and obsolete dictionary tokens are identified; definitions are not compared.

The standard dictionary name begins with a pound sign (#) and is expected to reside in the same directory. A dictionary update is only done when the `-update` flag is specified; otherwise, the dictionary is only checked, by default. Any tokens found in the standard dictionary that are missing from the dictionary will be added and those found in the dictionary that are missing from the standard dictionary will be removed.

`diskclear`
(Superuser only)

`diskclear [-|u]`

Marks each block of a server's database as invalid so that you can copy a new database to the disk. To clear the disk, select a server and type:
**`diskclear -`**

Do this prior to connecting a replacement server to a running system.

**Using this command erases the server's entire database.**

To reverse the effects of `diskclear`, type:
**`diskclear u`**

| Arguments | Description |
|---|---|
| - | Clear database |
| -+ | Writes to disk (cannot be repaired by running `diskclear u`) |
| -- | Read database blocks |
| -<number> | The number of blocks to process at one time |

| Arguments | Description |
|---|---|
| u | Unclear database |

`diskcopy`

```
diskcopy -[<# of simultaneous copies>]
diskcopy <start block> [<end block>]
diskcopy -s
```

Copies database from master computer (usually server A) to a replacement computer. Enter it on the replacement server.

Select the `-s` parameter to minimize impact on system performance.

`doc`

```
doc  -g[qu] <queue> [<title>...]
doc  -p[qu] <queue> [<file>...]
```

`title` must be one word or enclosed in quotes.

The `-g` option does not take any file name parameters; standard output is assumed. If output is to be written into a file, shell file redirection must be used (as in "> output-file-name"). The `-g` option defaults to plain text input / output and lines are terminated with CR/LF characters.

The `q` modifer specifies that the input / output will be in NSML format (not plain text). The `u` option specifies Linux line termination is used—that is, lines are terminated with an LF character.

The `-p` option will take input from its standard input if no files are specified. This will put a single story into the database composed of the data read from its standard input up to an end of file condition.

`ed`

```
ed <file pathname>
```

This command initiates the line editor used to edit text files. Since each server has its own copy of each site file, always select *all* servers before editing a site file. Procedures for using this line editor are covered in Appendix H. The line editor is a Linux-supplied program; refer to Linux documentation for more information.

`force`
(Superuser only)

```
force    -
force    [-q]    [<name>] ...
force    [-q]    "created>date1<date2" [<name>] ...
force    [-q]    "lastlog>date1<date2" [<name>] ...
force    [-q]    "passchg>date1<date2" [<name>] ...
```

*When using angled brackets (>,<)in a parameter, they must be enclosed within quotation marks. The same applies to characters, such as the exclamation point (!) and asterisk (*).*

This command forces users to change their password. For instance, to force user Harris to change her password, type:**`force harris`**

`grpcheck`

```
grpcheck [-v] <group story queue>
```

This command validates groups and aliases defined in each of the stories in the group directory (`SYSTEM.GROUP` by default). It then builds the alias file used by iNEWS.

**grpcheck Command Options**

| | |
|---|---|
| `-v` | Display processing status as grpcheck traverses the queue. |
| `-vv` | Display processing status as well as group, user, and alias statistics encountered. |
| `-vvv` | Display messages from -v and -vv, as well as the final list of groups and their members. |

`gtraits`
(Superuser only)

```
gtraits  add  <group name>
```

This command creates groups of users and modifies the security of existing groups.

The following lines show syntax for the `gtraits` command:

```
gtraits  changegroup  <pathname>
gtraits  delete  <group name>
gtraits  interactive
gtraits  list-
gtraits  list  [<group name>|<user name>]
gtraits  rename  <old group name>  <new group name>
gtraits  transfer  <source group name>  <destination group
name>
```

The first letter of each option can be used for shorthand.

`help`

```
help  <command name>
```

Displays information on how to use other commands. For instance, to get instructions for the `list` command, type:

**`help list`**

A message similar to the following appears:

```
usage:
   list configuration     [<termid> | <name>...]
   list directory         [<name>...]
   list terminal          [<termid> | <name>...]
   list user              [<name>...]
   list queue  <name>     [<record limit>]
```

`hogs`

```
hogs  [<pathname>]
```

Scans the directories or queues you specify and displays usage information for them. You can use this command to get an idea of which queues are consuming the most space and how much is being consumed. This command is most useful when used on the People directory. For instance, to display usage information for the People directory, type:

**`hogs people`**

To execute a hogs command and send the results to a system printer, use the following format:

`NRCS-A$` **`print <system printer number> hogs .`**

A display similar to the following appears:

```
%  USED   SHARED   HELD   LOCKED   PURGE   QUEUE NAME
0  136    20       0      0        0       PEOPLE.ARLIN
```

This command can take up to 12 hours to generate. The primary number to observe is the number under `USED`. If this number is much higher for a certain queue than the numbers for other queues, you can determine where material needs to be downsized.

As indicated by the pound sign (#) in the prompt, you must be in superuser mode to execute commands. Also, you must use 24-hour military time.

Below is an example of using the `at` command to execute a `hogs` command at a specified schedule, sending the results to a file (`hogs.report`), and then using the `doc` command to copy that file to a queue for later review.

```
NRCS-A# at 11:10
/exc/hogs . > /tmp/hogs.report
doc -pu people.sysadmin.notes /tmp/hogs.report
<Ctrl+D>
job 1001340698.a at Sat Jul 10 11:10:00 2004
NRCS-A#
```

`idiff`

`idiff <file1> <file2>`

This command allows you interactively compare two files and select and/or edit the portions of the files that differ and create a composite file. The program displays the differing portions of two files and gives you the option to put the portion from the first file or the second file into the output file.

You can also edit the display using the vi program and put the resulting portion into the output file, named `idiff.out`.

`list`

`list [mailbox=<# | name>] c [<terminal #> | <program name>] ...`

| | |
|---|---|
| list configuration | [<termid> \| <name>...] |
| list directory [- (a \| f \| g \| o \| s \| u \| v)] | [<name>...] |
| list group | <name \| {name}> |

| | |
|---|---|
| list queue [- (a \| b \| d \| f \| g \| m \| s \| v)] | <name> [<record limit>] |
| list session [- v \| p] | [<termid> \| <name> ...] |
| list user [- (h \| l \| m \| p \| r \| t \| v)] | [<name \| groupname>...] |
| list Blob [-v] | [<name>] |

**Configuration Options**

computer=<name>

mailbox=[<number> | <name>]

**Directory Options**

list ap=<number> d

list as=<number> d

list flags=<flags> d

*The* `flags=<flags>` *directory option allows the use of directory flags from the set* `SRP(p/l)opLIsUGQ(O/N/A)XWFi` *to be used to select the directory entries to be listed. To see all indexed directory entries use:* `list flags=i d`. *To see all sequential, read-only, and printable directory entries use:* `list flags=SRP d`.

list form=<name> d

*The* `form=<name>` *directory option matches the directory entry queue form and story form names.*

list mailbox=[<number | name>] d

list ng=<group> d

list purge=[<days>.<hours> | <hours>] d

list rg=<group> d

list rwg=<group> d

list rwng=<group> d

list save=[last | original | none | all] d

list sortfield=<field> d

list wg=<group> d

**Queue Options**

list qindex=<index value>

**User Options**

list blacklist=[b | -] user

list “created>date1<date2” user

list keyboard=<number> user

list “lastlog>date1<date2” user

list “passchg>date1<date2” user

list password= user

list readrate=<number> user

list session=<number> user

list simplified=[s | -] user

list su=[n | -] user

`list B`

Lists details of the "system blobs" stored on the server. System blobs are used to store bitmaps for the title-entry tool and for the simplified user interface settings. The following is a sample output:

```
# list B
 File-Id    Size  Date       Time       Name
     449    7287  Jul 10 00 07:27:19  BM000001
     457    7238  Jul 10 00 07:28:16  BM000002
   84435    6066  Nov 30 00 08:16:39  BM000003
   84363    6066  Nov 30 00 10:44:13  BM000004
   84403    1171  Nov 30 00 10:56:40  BM000005
   88956  265255  Jan  4 01 16:05:18  BM000006
  103749   15439  Mar 22 01 00:40:24  BM000007
     498   11212  Jan 14 00 15:35:28  BM000008
     530   15439  Jan 14 00 15:53:46  BM000009
     546   91416  Jan 26 00 09:47:30  BM000010
     638   14342  Apr  4 00 13:11:17  BM000011
     653    8608  Apr  4 00 13:18:09  BM000012
     662   26673  Apr  5 00 13:46:54  BM000013
  104423      40  Mar 21 01 15:16:28  SimplifiedUISettings
```

`list c`

Lists current configuration of the system.

**list c Command Options**

| |
|---|
| computer=<name> |
| mailbox=[<number> or <name>] |

`list d`

```
list  [<option>]  d-[a | f | g | o | s | u | v]
[<directory name>...]
```

**list d Command Options**

| | |
|---|---|
| d-a | Abstract printing traits |
| d-f | Queue and story form names |
| d-g | Group information |
| d-o | Order user |

| | |
|---|---|
| d-s | Son count and sequence number |
| d-u | Lock user |
| d-v | Verbose mode |

Lists information about the specified directory or queue. If no directory or queue name follows `d`, the command displays information on the entire database.

For instance:

```
# list d dead
 SRPlo-LIsUGQSXWFiT sortfield    purge dis mbox directory
Q-R-----I--G--X---- TITLE        P3.0  D1  -    DEAD
# list d-f dead
 SRPlo-LIsUGQSXWFiT queue form story form directory
Q-R-----I--G--X----                    DEAD
# list d-v dead
 SRPlo-LIsUGQSXWFiT sortfield    purge ap,  al, as   dis mbox
DEAD:
Q-R-----I--G--X---- TITLE        P3.0  A000,000,000  D1  -
     rg=-                    wg=-                    ng=-
     queue form=                        story form=
```

`list g` — `list g [<user or group name>] ...`

Lists group information.

`list q` — `list [<option>] q-[a | b | d | f | g | m | s | v] <name> [<limit>]`

**list q Command Options**

| | |
|---|---|
| q-a | Record address |
| q-b | Reverse order |
| q-d | Include deleted entries |
| q-f | Story form |
| q-g | Read-and-write group information |

| | |
|---|---|
| q-m | Who moved, duplicated, or killed the queue |
| q-s | Queue stamp |
| q-v | Verbose output |

Lists information on the contents of a queue. For instance:

NRCS-B$ **list q people.palmer.new 1**

A display similar to the following appears:

```
PEOPLE.PALMER.NEW id=126126
rec  quick index    LHDM-WObfpRmF  f.id   time modified-time
25   h-disd         ---M---------  13735  1    May2 17:07 2000
```

The index value consists of the selected sort field of the story you want to list. The quick index (`qindex`) value is optional, but must be a single word, and is not case-sensitive.

For instance, to get information for a story called “Nomad” in the queue `PEOPLE.SMITH.NOTES`, type:

**list qindex=nomad q people.smith.notes**

`list s` `list s -[p | v] [<session id> | <name>]...`

Lists session information, such as users currently logged in.

The `-p` option causes the process id for the controlling session process to be printed. The `-v` option causes the IP address for the session to be printed if the session is a network session, such as an iNEWS Workstation.

`list u` `list [<option>] u[-(h | l | m | p | t | r | v)] [<user or group name>]...`

Lists user traits information, such as read rate, the keyboard description story, system setup and preferences, and mail and home queues.

If no name follows `u`, the command displays information about all users; otherwise, it displays information about the listed user, such as:

```
NRCS-A$ list u-v danielmi
```

The verbose result of the command will look something like this:

```
user      rr   kb   su  m   SOEKCVTH  sc  queues

danielmi  180  0    n   i   -OEKCVTH  sc  dest: PEOPLE.D.DANIELMI.NOTES
                                           home: PEOPLE.D.DANIELMI
                                           mail: PEOPLE.D.DANIELMI.MAIL

NRCS-A$
```

The flags (`rr kb su m SOEKCVTH sc`) in the header provide current status information. The flags are:

| | | | |
|---|---|---|---|
| rr | Readrate | K | Can Kill All |
| kb | Keyboard | C | Can Connect |
| su | Superuser | V | Can Video Browse |
| m | Insert/Overstrike Mode | T | Can Technical Direct |
| S | Simplified User | H | Can Highlight Read |
| O | Can Order | s | Can Configure Shortcut Toolbar |
| E | Can Enter & Remove | c | Can Configure Colors |

The letters in the header are defined as follows:

| | |
|---|---|
| u-h | Home queue |
| u-l | Last login |
| u-m | Mail queue |
| u-p | Preferences |
| u-r | Real name |
| u-t | Tracking information |

| | |
|---|---|
| u-v | Verbose output |
| <option> | list blacklist=[b \| -] user |
| | list “created>date1<date2” user |
| | list keyboard=<number> user |
| | list “lastlog>date1<date2” user |
| | list “passchg>date1<date2” user |
| | list password= user |
| | list readrate=<number> user |
| | list session=<number> user |
| | list simplified=[s \| -] user |
| | list su=[n \| -] user |

*When using angled brackets (>,<)in an option, as shown above, they must be enclosed within quotation marks. The same applies to characters, such as the exclamation point (!) and asterisk (*), as shown below.*

```
list user “a*”
```

Lists all users whose login names begin with the letter ‘a’.

`logout`

```
logout  [<device #>] ...
logout  all
```

Logs out a workstation. When you use `logout`, it saves the user’s work before logging out his or her workstation. This command does not log out users in a connect session.

To log out specific workstations, follow the `logout` command with the device numbers of the workstations you want to log out. For instance, to log out workstations 12, 34, and 91, type:

**`logout 12 34 91`**

To log out all workstations, use `logout all`. Before logging users out, always broadcast a warning message and give them a chance to log out on their own.

`makeccutab`
(Superuser only)

`makeccutab -i[sv]`

Builds new dictionary translations into PCU programs.

To use this command, stop all PCUs, then become superuser and type:
**`makeccutab -i`**

*For testing, the command format is:* `makeccutab -f [sv] filename`.

**makeccutab Command Options**

| | |
|---|---|
| s | Standard translations |
| v | Verbose output |
| vv | Very verbose: show progress plus translations |

*The dictionary definitions are installed into the* `/exc/ccu/at/printer` *file.*

`makemctab`

The file into which the dictionary definitions are installed are `/exc/monitor`, all configured mcs drivers resident in the `/exc/ccu/at` directory, `/exc/ccu/at/cmct`, and all configured mcspc drivers resident in the `/exc/ccu/tftp/mcspc` directory.

The usage for the `makemctab` is the same as the `makeccutab`.

`makeshift`
(Superuser only)

`makeshift -[v|i|a|p|r|f <shift-file>] file1 file2 ...`

Manages the case-shifting dictionary that iNEWS uses to determine how to convert lowercase characters to their uppercase counterparts, and vice-versa.

Use the `makeshift` command in maintenance mode when you install iNEWS to implement the case-shifting dictionary appropriate for the national language used at your site.

### makeshift Command Options

| | |
|---|---|
| v | Verbose. To diagnose the case-shifting dictionary for potential errors, displaying messages for each line in the file. Checks that the file is readable and contains shift tables. |
| i | Install. To install the shift table into files you specify in the filename list. |
| a | Ask. To confirm installation of each file as you are installing the shift tables. Forces installation. |
| p | Print. To print shift tables contained in each file you specify in the filename list, with formatting similar to that in the default case-shifting dictionary. This option does not build or install the shift tables. |
| f | File. Specify with a <shift-file> filename, to use a file other than the default file for the case-shifting dictionary. The default file is (`/site/dict/shift`). |
| r | Recursive. (For directories.) |
| <shift-file> | When you specify the -f option, enter the case-shifting dictionary name you want to use instead of the `/site/dict/shift` file. |
| <file1> <file2> ... | When you specify the -i or -p option, enter one or more file names to install or print. If you specify a directory path instead of a file name, the `makeshift` program processes each file in the directory, then returns to the original directory. |
| [path] | When processing files in a directory, the `makeshift` program ignores additional directory pathnames it encounters, rather than recursively scanning child directories. To have `makeshift` scan all files in a directory, specify the directory path in the filename list. |

`maketab`
(Superuser only)

`maketab -i[sv]`

**maketab Command Options**

| | |
|---|---|
| s | Standard translations |
| v | Verbose output |

Use this command after making changes to the dictionaries (or before the system is connected) to build the new translations into programs. To use this command, first stop all PCUs. Then become a superuser and type:
**`maketab -i`**

The usage for the `maketab` is the same as the `makeccutab`.

*For maketab, the dictionary definitions are installed into the following programs:* `/exc/news`, `/exc/gnews`, `/exc/snews`, `/exc/nxserver`, *and* `/exc/newsmail`. *Day and month abbreviation definitions are installed into:* `/exc/news`, `/exc/gnews`, `/exc/snews`, `/exc/nxserver`, `/exc/newsmail`, `/exc/telex`, `/exc/rxnet`, *and* `/exc/server`.

`msgclean`

`msgclean -[r|d|o|t] [<username>|<number of days>]`

**msgclean Command Options**

| | |
|---|---|
| -r | Remove invalid messages |
| -d | Show messages that are removed |
| -o | Show outstanding messages |
| -t | Tabulate outstanding messages |
| <username> | Specified user (use “*” for wildcard match) |
| “#<days>” | Older than number of days |

*When using characters, such as angled brackets (>,<), the pound sign (#), exclamation point (!) and asterisk (*), they must be enclosed within quotation marks.*

If you remove a user who has unread messages, those messages remain in the database. You can remove them using the `msgclean` command followed by `-rd`. For instance, to remove the pending messages for all users that have been removed from the system, type:

**`msgclean -rd "*"`**

You can substitute the name of a removed user for the "*" to remove only the messages for that user.

`offline` `offline [silent]`

Puts iNEWS offline. Users cannot log in, but users already on the system can continue normal function. The `silent`option suppresses diagnostics for network connections that are refused because the system is offline. Both `nxserver` and `workserver` typically generate these diagnostics.

The datastamp generator, server timer, includes an indication that the system is offline in the timestamp messages produced on the console. This provides feedback that the system is offline when the `silent`option was specified.

The `status` command will print "OFFLINE (silent)" when the system is offline with the `silent` option turned on.

You can keep the system offline and turn off the `silent` option by reentering the `offline` command without the `silent` option.

`online` `online`

Puts iNEWS online. Users can log in and use the system.

`otod` `otod <number> …`

**otod Command Options**

| | |
|---|---|
| leading 0 | Octal |
| leading 0x | Hex |
| leading = | Next characters (as characters) |

| | |
|---|---|
| leading - | Negative number |
| leading _ | Two-character compose sequence |
| leading % | strftime format - subsequent numbers formatted with format for local and gmt times |
| leading L | locale to use for date formats |

This command converts numbers from one base (such as decimal) to another (such as octal). For instance, to convert decimal 32, type:

**otod 32**

A message similar to the following appears:

```
h(0x20) o(040) d(32) u(32) SP
32.0.0.0 Wed Dec 31 16:00:32 1969
```

In this listing, `h` stands for hexadecimal, `o` for octal, `d` for decimal, and `u` for unsigned decimal. The number conversions are followed by the corresponding ASCII character (space, in this case), and the date value. The `32.0.0.0` is the number shown in IP address format.

The `%` and `L` options can assist you in handling different locale settings and formats. The following are two examples:

Type: **otod %X 32**

A message similar to the following appears:

```
h(0x20) o(040) d(32) u(32) SP
0.0.0.32  Wed Dec 31 18:00:32 1969
lcl: 18:00:32
gmt: 00:00:32
```

Type: **otod Lfr 32**

A message, with the date in French, similar to the following appears:

```
locale is fr
h(0x20) o(040) d(32) u(32) SP
0.0.0.32  mercredi 31 décembre 1969, 18:00:32
```

print

```
print  <printer #>  <command>
```

Redirects the output from any console command into a story in the Dead queue and prints it.

For instance, to print the configuration file on printer 1, type:

**`print 1 cat /site/config`**

query

```
query  <pcu #> | <mcs pc devid>
query  <device #>  <qtype>  [parameters]
```

Syntax:

```
<devid>  ioregister
<devid>  iocontrol
<devid>  proc
<devid>  input
<devid>  output
<devid>  read  <addr> <length>
<devid>  absread  <seg> <addr> <length>
<devid>  writeb  <addr> <byte value in hex>
<devid>  writew  <addr> <word value in hex>
<devid>  abswrite  <seg> <addr> <word value in hex>
<devid>  netstat
<devid>  bstat
```

Tests communication between a PCU or MCS-PC and its server. Type **`query`** followed by the device number of the PCU or MCS-PC. For instance, to test PCU 20, type:

`NRCS-A$` **`query 20`**

A message similar to the following appears if the PCU or MCS-PC is functioning properly:

```
#(@) 1.5.1.12 DOS PCUOS OK
```

If the system returns an OK message a few seconds after you execute the command, the PCU or MCS-PC is functioning properly. If there is a problem, the server displays a failed message after about a minute.

`reconnect`

```
reconnect  <name>  [<option>=<value>] …
```

The options for this command are the same as for the `connect` command.

**Enter the correct identifier, such as A, B, C,or D, for the host computer (server). The `reconnect` command must be entered on all of the currently connected servers as well as the server to be reconnected. The`reconnect` command will exit with a diagnostic if the specified server is already connected.**

Connects a server to a running iNEWS newsroom computer system. You must type the `diskclear` command first on the server being reconnected before reconnecting it to the iNEWS newsroom computer system.

`rename`
(Superuser only)

```
rename  [- u|v|r]  <old name>  <new name>
```

**rename Command Options**

| | |
|---|---|
| -u | Update user records only, changing mail, destination, and home directory entries that match <old name> |
| -v | Verbose output |
| -r | Resume interrupted operation |

**System must be offline. Resuming interrupted operations after changing queue or directory names could cause loss of data.**

Renames any directory or queue in the database. You must become superuser, take the system offline, and log out all users before using `rename`.When you use this command, the system must be named and offline. For instance, to rename the People directory to `PEOPLE.STAFF`, type:

```
rename people people.staff
```

The command creates any new directory levels that are necessary.

`reorder`

```
reorder <parent> <child> <position>
```

This allows you to change the order of a directory by putting one of the child entries into a new position in the directory. Reordering is only done for Sequential directories.

The `<position>` parameter can be:

```
-a <sibling>  (place <child> after <sibling>)
-b <sibling>  (place <child> before <sibling>)
-<number>     (move <child> up <number> positions)
+<number>     (move <child> down <number> positions)
<number>      (place <child> at <number> position)
```

The `<child>` and `<sibling>` parameters do not include the full pathname; only the last level name relative to the parent name.

`restart`

```
restart  [-v]  <device> | all
```

**If you restart a device when a user is editing, data could be lost.**

Restarts one or more devices, an entire PCU (and its devices), or every PCU and their devices on a server. The `restart` command stops and reloads the currently executing program(s).

To restart a device, type **`restart`** followed by the device number. For instance, to restart printer 41, type:

```
restart 41
```

To restart all devices on a PCU, type **`restart`** followed by the PCU's device number. To restart all devices on your system, type:

```
restart all.
```

After each restart, you see a `Hot-to-go` message for each device as it starts. If the device does not start, you see a message indicating that the restart of that device has failed.

`searchtape`

```
searchtape [on <device>] [from <date> to <date>]
[<max # stories>] for <word> …
searchtape [on <device>] [just <date>]
[<max # stories>] for <word> …
```

Use one of the following date formats: YY, YYMM, YYMMDD, or CCYYMMDD.

Searches a tape created by the `dbdump` command and recovers stories from it. Stories that contain a word specified in your search are restored to the queue `SYSTEM.SEARCHTAPE`. For instance, to search a tape for stories containing the word "dinosaur," type:

```
searchtape for dinosaur
```

A message similar to the following appears:

```
8 stories restored to SYSTEM.SEARCHTAPE
```

In this case, `searchtape` found eight stories containing the word "dinosaur" and placed them in `SYSTEM.SEARCHTAPE`.

When searching a tape, you can specify a date range when the story was saved, as well as the maximum number of stories for the system to restore.

`send`

```
send  <username>  "<message>"
```

Lets you send instant and intersystem messages to users from the console. Messages can be sent to the special user—`computer`—on another system by using the username, `computer@other-system`. These messages are printed on the console of the other system.

Here are two examples of messages sent to user Smith:

```
send smith "are you editing ap-arson"
send smith "please log out now"
```

`shutdown` — `shutdown`

**Typing this command while users are editing may cause data loss.**

Stops all devices and closes the database. It is used before halting the system. To use this command, you must first select *all* servers and log everyone off the system.

`sitedump` (Superuser only) — `sitedump [-d<device> | -f<file>] [-ehv]`

Makes backups of your system's site files.

**sitedump Command Options**

| | |
|---|---|
| -d | `<device>` is the name of a tape device. The actual device name used will be `/dev/<device>`. The default is `/dev/tape`. |
| -f | `<file>` is the name of a file to use as the archive. |
| -e | Eject tape when done. |
| -h | Help - show usage information. |
| -v | Verbose |

`siterestore` (Superuser only) — `siterestore [-d<device> | -f<file>] [-ehtv] [<file> ...]`

**All site files will be replaced by the version on the backup tape. You will lose the version currently on your system.**

Restores site files and programs backed up to tape with `sitedump`.

*After performing a siterestore, run the* `grpcheck` *command to rebuild the mail aliases file.*

**siterestore Command Options**

| | |
|---|---|
| -d | `<device>` is the name of a tape device. The actual device name used will be `/dev/<device>`. The default is `/dev/tape`. |
| -f | `<file>` is the name of a file to use as the archive. |

| | |
|---|---|
| -e | Eject tape when done. |
| -h | Help - show usage information. |
| -t | Show table of contents; no files are restored. |
| -v | Verbose |
| <file> ... | Only the named files are restored. If a directory is specified, all files in the directory will be restored. |

`softdump` `softdump [-d <device>] [-ehv]`

Dumps a complete image of the software area of the disk to tape.

*You must be a superuser to backup the root disk.*

With the exception of `-f`, the options for `softdump` are the same as the `sitedump` command.

`softrestore` `softrestore [-d <device>] [-ehtv] [<pattern>...]`

**This function must be performed from single-user mode. Restoring an old or inaccurate version of the data may cause functionality problems.**

Use this command only when advised to do so by Avid personnel to restore the software area of the disk from a `softdump` tape.

*You must be a superuser to restore the root disk.*

### softrestore Command Options

| | |
|---|---|
| -d | `<device>` is the name of a tape device. The actual device name used will be `/dev/<device>`. The default is `/dev/tape`. |
| -e | Eject tape when done. |
| -h | Help - show usage information. |
| -t | Show table of contents; no files are created. |

| | |
|---|---|
| -v | Verbose |
| <pattern> ... | Select only files with names that match pattern. |

`startup` `startup`

Starts the system's devices after they have been shut down. The system must be offline and all devices must be stopped.

`status` `status [all | license]`

Displays system connection information: which server is the master, if system is running single or dual, and the disk status.

There are three disk statuses:

- **OK** - When the system is up and running either as dual or single.
- **Cleared** - When you have cleared the database of a failed server.
- **Unknown** - When you reconnect the CPUs following a `diskclear`. When the `diskcopy` procedure has completed and the database has been mirrored, the disk status will change back to OK.

To list basic system information, type:
**`status`**

A message similar to the following appears:

```
A is online  ID is NRCS
System is AB. Master is A.
Disk status is UNKNOWN
```

To list system options set in the system profile, type:
**`status all`**

To list the system's license information, type:
**`status license`**

`stop` `stop [-v] all | <device number>`

**If you stop a device when a user is editing on the device, data could be lost.**

Stops activity on a server prior to shutting it down. Before using the `stop` command to stop an activity, use the `broadcast` command to notify users the system will be going down, and log out all devices on the affected servers. To stop all devices on a server, use `stop all`.

To stop a device, follow `stop` with the number of the device you want to stop. For instance, to stop workstation 12, type:

**`stop 12`**

The `stop all` command differs from the `shutdown` command in that the free list remains in memory and is not flushed back to the disk.

`su` `su`

This Linux command allows you to enter superuser mode. Type **`su`**, then type the superuser password when prompted.

The display looks similar to the following:

```
NRCS-A$ su
password:
SU: so /dev/console
```

`unbusy` `unbusy [-i] <queue name>`

Removes edit and order locks from the specified queue in your database.

The `-i` option allows you to ignore the inverted database trait.

**If a user is actually working in the file, removing the lock could cause data loss.**

`utraits`
(Superuser only)

`utraits <username> [<option> <value>] [+|- flag]`

**utraits Command Options**

| | |
|---|---|
| destination | realname |
| editmode | su |
| keyboard | home |
| readrate | password |
| mail | |

*The* `bloblist` *option is useful for removing iNEWS Workstation user preferences should they become corrupted.*

**utraits Command Flags**

| | | | |
|---|---|---|---|
| b | Blacklist | ka | Kill all |
| bloblist | Remove blob list | log | Log client/server communication |
| c | Connect | o | Order |
| cc | Configure colors | s | simplified user interface |
| cs | Configure shortcut toolbar | td | Technical direct |
| er | Enter and remove | vb | Video browse |
| hr | Highlight read | | |

Sets a user's traits. For instance, to assign keyboard 12 to user Jones, type:
**`utraits jones keyboard 12`**

The `<username>` can be "all", a pattern, or a user name. You can enclose the real name in quotation marks to allow for embedded spaces. Use `"<string>*"` to indicate wildcards. For instance, "a*" applies to all users with names starting with the letter A.

`version`

```
version [-<alternate pattern>] <filename> …
version [+]
```

Displays the version and platform of the iNEWS software you are using. Type:

**`version`**

A message similar to the following appears and displays product version information:

```
(c) 2004
This computer program is protected by copyright law and
international treaties. Unauthorized reproduction or
distribution of this program, or any portion of it, may
result in severe civil and criminal penalities.


U.S. GOVERNMENT USERS RESTRICTED RIGHTS: Use, duplication
or disclosure by the U.S. Government is subject to
restrictions as set forth in subparagraph (c)(1) of FAR
clause 52.227-19, COMMERCIAL COMPUTER SOFTWARE -
RESTRICTED RIGHTS or, in the case of the Department of
Defense or its subcontractor, is subject to DFARS
227.7202-3, RIGHTS IN COMMERCIAL COMPUTER SOFTWARE OR
COMMERCIAL COMPUTER SOFTWARE DOCUMENTATION.


version: 2.0.0.21 LINUX
```

You can also use this command to display a particular program's version number. Type **`version`** followed by the program's name. For instance, to find out which version of the `dbsort` program you are using, type:

**`version dbsort`**

A message similar to the following appears:

```
dbsort: 2.0.0.21 LINUX
```

`wholockedit` `wholockedit <queue name> [all]`

Displays who locked a story. For instance, to find out who locked a story in `PEOPLE.SMITH`, type:

**`wholockedit people.smith`**

To find out who last modified each story in a queue, type **`wholockedit`** followed by the name of the queue and the keyword **`all`**.

`wiredump` `wiredump <device number> <speed> <bits> <file> <count>`

or

`wiredump <device number> <speed> <bits> -<code[rotate]> <style>`

| | |
|---|---|
| <bits> | Bits per character (5, 6, 7, or 8) or speed options, as interpreted by the `ccuspeed` program. |
| <file> | File name for collected data |
| <count> | Number of bytes of data to collect |
| <code> | b for Baudot, t for TTS, a for ASCII, w for Wire ASCII, or e for 8-bit ASCII |
| [rotate] | 1-7 to rotate right <x> bits; 8 for one's complement (010=>101); 9 for inversion (123=>321) |

If a problem exists with your wires, you may need to use this command to collect raw wire data so it can be sent to iNEWS for testing. The `wiredump` command can also be used for debugging format changes made by a wire service. A wire dump can be sent to the screen or to a disk file.

For instance, if device 35 is a 1200 baud, ASCII, 7-bit wire, type the following to print the first 200 characters to the screen:

**`wiredump 35 1200 7 -a c`**

To dump wire data to a file, use the following format. The file name refers to the output file into which the `wiredump` data is placed. It should include the name of the wire and be preceded by the `/tmp` directory name. The count is measured in number of bytes.

For instance, if device 35 is the APTV wire, to capture the first 2000 characters, type:

**`wiredump 35 1200 /tmp/aptv.raw 2000`**

A display similar to the following appears:

```
loadtime = 0:11
NRCS-A$ restart 35
W35: 14:34:09 Hot-to-go
```

# Job List Commands

The following section provides a list of commands that can be used in a job list, which is created and modified in the database. The command's format and description are provided, followed by a list of servers that can utilize the command in their job lists.

`at`

`at <hh:mm>`

Specifies the time of day when a task will take place. You can combine this instruction with the keyword `on` to specify both the day and time for the task. Applies to action and tx servers.

`bpoll`

`bpoll <queue name> [<polling interval> [<polling duration>]]`

Works like `poll`, except it reads stories in the primary queue in reverse direction. If the polling interval and duration are not specified, a single scan of the primary queue will be done. Applies to timed-interval action and tx servers.

`bscan` — `bscan <queue name> [priority | all | everyentry]`

Works like `scan`, except it reads stories in the scan queue in reverse chronological order, reading the newest stories first. This is convenient for very large queues. Applies to action and tx servers.

The `priority` option forces the action/txnet server to interrupt the scan of another queue if the action/txnet server receives a mailbox notification. The `all` option forces the action/txnet server to scan the entire queue instead of the limit of 10 stories when it has more than one queue to scan. The `everyentry` option forces the server to process each entry in a queue, not just modified entries.

`distribution` — `distribution <distribution story queue> [<error queue>]`

Specifies the queue containing the distribution story and, optionally, an error queue for stories whose distribution codes cannot be processed. Applies to distribution and parallel servers.

`dup` — `dup <destination queue> [<distribution code>]`

Copies the stories in the scan queue to a queue you specify, optionally including a distribution code with them. Applies to action and tx servers.

`every` — `every <dd.hh>`

Specifies the interval at which a task is performed. You can set this value in days and/or hours. Applies to action and tx servers.

`extension` — `extension <file extension>`

Applies to tx servers. When transmitting stories using HTML format, each transmitted story has a filename composed of a hexadecimal representation of the entry's qstamp and a filename extension of `.html`. Use the `extension` command to specify a different extension. Do not include the period in the `<file extension>` parameter. The defined extension will also be used in place of `html` for the published file. See the `publish` job list command for more information.

`ignore` `ignore [yes | no]`

Including `ignore yes` in a job list that performs validation ensures the server accepts any values for the fields it is validating. The default is `ignore no`. Applies to action and tx servers.

`ignore-del` `ignore-del`

Causes a server to take no action when a story is deleted from its scan queue. Applies to action and tx servers.

*This option is not reset for each* `scan/bscan` *command set. Use* `send-del` *to restore processing of deleted queue entries.*

`local` `local <queue name>`

Specifies the primary wire queue. Applies to parallel servers.

`mailto` `mailto <recipient>...`

Mails the story as an e-mail text message to each recipient. The list of recipients is a space-separated list. If the `sendform` option is on, the content of fields in the story is included at the start of the e-mail text message. Only fields present in the story form assigned to the story are included. Each field is identified by the label text associated with the field in the story form assigned to the story.

`move` `move <destination queue> [<distribution code>]`

Moves stories from the scan queue to a queue you specify, optionally adding a distribution code to them. It must be the last instruction in a job list task. Applies to action and tx servers.

`number` `number <form field> <length> <error queue>`

Assigns a unique number to each story as the story is processed. Specify the form field that will contain the number and the number of digits for the number. Applies to action and tx servers.

`on` — `on <day> ...`

Indicates on which days of the week a time-interval task will occur. You can combine this instruction with the `at` keyword to indicate both day and time. Applies to action and tx servers.

`open` — `open <computer> <username> [<format> [<queue name> [<story name>]]]`

Initiates a network connection to a remote system for story transfer. The `username` you specify must exist with identical passwords on both the local and remote systems. Applies to tx servers.

The `<computer>` can include a port number. The format is `host:port`. If no `:port` is included in the `<computer>` parameter, the port defined by the rxnet service is used, if defined; otherwise, the port defined by the FTP service is used.

The `<username>` can specify a simple name or a name in the format `name@host`. Only the name portion is used to locally look up a password. The entire username is sent in the FTP `USER` command. This may allow connections to be made through proxy servers.

The format, if specified, must be one of the following: `2nsml`, `nsml`, `sep`, `or html`.

The queue name and story name are only used when the format is set to `HTML`. The queue name and story name are used to get the Web publishing template that controls the formatting of the story into an HTML page. If the queue name is not specified, the template is taken from the `SYSTEM.WEBFORMS` queue. The story name can be used to select a specific template from the queue. If not specified, the first story in the queue is used.

`order` — `order [yes | no]`

Indicates that order changes in the scan queue should be transmitted to the remote system. For this to work correctly, the destination queue on the remote system must have its update trait turned on.

The `order` command applies to tx servers and action servers. For action servers, the order of the scan queue is propogated to each of the `dup`, `move`, and `replace` command destination queues. Specifying order without yes or no is the same as `order yes`.

`poll`

```
poll <queue name> [<polling interval> [<polling duration>]]
```

Reads stories in the specified primary queue in a forward direction at certain intervals for a specified duration. Polling is used in conjunction with the `put` or `dup` commands for rundown mirroring. Stories with modified times greater than the time of the last scan are processed. If an interval and duration are not specified, a single scan of the primary queue will be done. Applies to timed-interval action and tx servers. See Chapter 13 of the *iNEWS Newsroom Computer System Setup and Configuration Manual* for more information.

`publish`

```
publish [no|yes]
```

When placed following a `scan` or `bscan` line, the txnet publishing option, `publish no`, disables appending information to the `PUBLISHED.HTML` file on the remote system when using HTML export. The default is set to `yes`. Applies to tx servers.

`put`

```
put  [<queue name>]
```

Sends stories over a Tx link to a specified queue on the remote system. Applies to tx servers.

The `<queue name>` parameter is optional. When it is not included, the story is put into the default destination queue of the user on the remote system.

`quiet`

```
quiet  [no|yes]
```

Including `quiet no` in a job list that performs validation sends a message whenever the server successfully validates a story. The default, `quiet yes`, means a message is sent only when a story fails validation. Applies to action and tx servers.

`remote` `remote <queue name>`

Identifies the secondary wire queue. Applies to parallel wire servers.

`remove` `remove`

Deletes stories from the scan queue. It must be the last instruction in a job list task. Applies to action and tx servers.

`replace` `replace <destination queue name> [<distribution code>]`

Works like the `dup` command, except that it updates stories in the destination queue only when they are already present in the destination queue. It does not add new stories to the destination queue. Applies to action and tx servers.

`scan` `scan <queue name> [priority | all | everyentry]`

Specifies the queue monitored by this task. The scan line must come before any instructions that manipulate stories in the queue, like `dup` or `move`. Ten stories are scanned at a time from each scan queue; adding `priority` to a scan line means all new or modified stories in that queue are scanned at once. The queue identified in the `scan` command as the priority queue is always the next queue in the multiple-scan job list, so if it is idle, other queues are processed. The system checks after every queue to see if new stories are ready for processing on the queue identified in the `scan` command. Applies to the action and tx servers.

The `priority` option forces the server to interrupt the scan of another queue if the server receives a mailbox notification. The `all` option forces the server to scan the entire queue instead of the limit of 10 stories when there are multiple queues to scan. The `everyentry` option forces the server to process each entry in a queue, not just modified entries.

*Deleted entries are still controlled by the* `send-del` *option.*

`send-del` `send-del`

Instructs the server to process story deletions in the scan queue; this is the default behavior. Use `ignore-del` to have the server take no action when a story is deleted from a scan queue. Applies to action and tx servers.

`sendform` `sendform`

Instructs the Tx link to transmit the full form text of each story, rather than just the story's form name. Applies to tx servers.

`source` `source <queue name>`

Specifies a queue that a distribution or keyword server should check each time it wakes up. Each task in a job list for such a server must begin with a `source` line. Applies to the distribution and keyword servers.

`validate` `validate <validation queue> <error queue>`

Activates form field validation for a server. It must include the queue name containing the validation story and an error queue for stories that cannot be validated. Applies to action and tx servers.

# Dialog Commands

This section describes the dialog commands. Some of these commands are equivalent to those available to a user during a connect session, while others are unique to dialogs.

Each command must begin on a new line and can be uppercase or lowercase. The system does not check dialogs for errors.

Many of these commands take one or more strings as their parameters. These strings are text and can be entered from the keyboard or, if you need to include characters not available from your keyboard, use their aliases or decimal equivalents. This applies to nonprinting characters, such as Escape and Enter, which are represented by `<esc>` and `<cr>` respectively.

Depending on which character set is used by the device to which you connect, you may need to translate certain keyboard characters to characters understood by the device. You may also need to translate some of the characters sent by the device to characters usable by your system.

Translating characters is called *mapping*, and there are three commands (`map`, `mapin`, and `mapout`) that allow you to do this. While these commands are also available to users, system administrators will usually use them in dialogs to set up the translations for users rather than leave it for users to do after they have connected.

`capture` `capture <destination queue name>`

Places a copy of all text received from the remote connection during a session in a story into the queue you specify.

Usually you invoke `capture` from the `cmd>` prompt. However, to turn `capture` on in a dialog, place the `capture` command and destination queue name at the point in the dialog where you want to begin capturing material.

You must include a destination queue unless you are restarting `capture` after having paused it. If you have not paused `capture` earlier in the dialog, leaving out the queue name generates an error and terminates the dialog.

`delay` `delay <# of seconds>`

Pauses the dialog for a number of seconds.

When the specified time has passed, the dialog resumes. Put the command where you want the dialog to pause and follow it with the number of seconds you want the dialog to pause. For instance, to pause the dialog for five seconds, type **`delay 5`**.

Although the dialog is suspended while this command is in effect, you can use the `quit connect` command to close the connection.

`diag` `diag [on | off]`

Normally, a dialog's diagnostic mode is off and screen output is suppressed while the dialog is running. However, you can use the `diag` command to turn the dialog's diagnostic mode on so you can see what the dialog is doing as it executes.

Usually you want the diagnostic mode on only when you are debugging a dialog, so you can determine exactly where any errors occur. Place a `diag on` command in the dialog at the point where you want to start debugging.

To see what happens during just one part of a dialog, bracket that portion of the dialog with `diag on` and `diag off` commands.

`echo` `echo [on | off]`

Turns local echo on or off.

Turn on local echo in any dialog used to connect to a device or information service that does not echo back what the user enters. This way, the user can see what he or she is entering.

To turn on local echo, place `echo on` in the dialog where you want local echo turned on. Use this command at or near the beginning of the dialog.

While you can turn local echo off using the `echo off` command, you are not required to do so. Local echo is turned off automatically when the dialog finishes.

`escape` `escape <escape character>`

To change the escape character (used to bring up the `cmd>` prompt) from within a dialog, use the `escape` command.

For instance, to change the escape character from the default Ctrl+] to Ctrl+Z, use the `escape` command. Represent the Ctrl+Z character as `<26>` (its decimal value).

*Do not change the escape character to Ctrl+A, Ctrl+Q, or Ctrl+S. These characters have other important functions.*

The escape character is reset to the default (Ctrl+]) when the user closes the connection.

`expect`

```
expect
<delimiter><string1><delimiter><string2><delimiter>...
```

Instructs the dialog to wait for the device to which the service has connected to send a string (*`string1`*). If that string is not received within five seconds, `expect` sends the second string (*`string2`*) to the device and waits for a third string (*`string3`*). This continues until an expected string is received within the five-second limit or `expect` runs out of strings.

To use the `expect` command, follow it with the character—that is, `<delimiter>`—you want to use as the delimiter to separate each string in the list. The delimiter can be any character. Follow this with the first string you want `expect` to wait for. Then add a delimiter and the string you want `expect` to send if it does not receive the first string. You can add as many strings as you want.

For instance, some devices may not display a `login:` prompt unless you press Enter. To have the dialog send a carriage return to the device if it does not receive the `login:` prompt immediately, use the `expect` command. If the first character following `expect` is a comma, this sets the comma as the delimiter used to separate strings following the command from each other.

If you do not place a string between two delimiters, this indicates a null string. If you have the `expect` command wait for a null string, it considers any string it receives to be a match. If you have the `expect` send a null string, it does not send anything, but instead waits for the next expected string.

Although the dialog is suspended while this command is in effect, the user can employ the `quit connect` command to close the connection.

`heol`

```
heol [on|off]
```

If necessary, you can have the system insert a hard end-of-line character (HEOL) after each line of captured text.

Put the `heol on` command in the dialog at the point where you want this feature turned on.

To turn off this feature, use `heol off`.

If you are calling a device that contains information in tables or columns, have the system insert an HEOL at the end of each captured line. This way, tables and columns you capture retain their format. Put the `heol on` command in the dialog.

`map` `map <local character> <remote character>`

Translates a character (local character) entered at the keyboard to some other character (remote character) before sending it to the device to which you are connected. Likewise, if the system receives a remote character from the device, the `map` command translates it to local character before sending it to a workstation.

`mapin` `mapin <remote character> <local character>`

Translates a specific character (remote character) received from the device to which you are connected to another character (local character).

This translation affects only characters received from the device to which you are connected. It has no effect on the character when you enter it.

`mapout` `mapout <local character> <remote character>`

Translates a particular character entered at the workstation (local character) to another character (remote character) before it is sent to the device to which you are connected.

For instance, some devices use a limited character set that does not recognize lowercase letters. To connect to such a device, you would want to map all lowercase characters to their uppercase equivalents. For instance, to map a to A, use this `mapout` command:

**`mapout a A`**

This has no effect on characters received from the device. Only characters typed at the workstation are translated.

`message` `message <string>`

Sends the string to the screen.

This command informs users that the system is active and functioning during a dialing session.

`pass` `pass <resume character>`

Suspends a dialog and yields control to the user.

Whatever the user enters after the `pass` command is sent directly to the device to which the user is connected. When the user enters the character defined in <`resume character`> parameter, `pass` resumes the dialog.

To use `pass`, place it in the dialog where you want to yield control to the user and follow it with a character you want the user to enter to resume the dialog. When the user enters this character, `pass` sends it to the device and then resumes the dialog, preventing further user input.

For instance, suppose you have a dialog that logs you into an information service. For security reasons, you want the dialog to pause at the `password` prompt, let you enter the password, and then resume. Do this using the `pass` command followed by a `<cr>` so pressing Enter after entering the password resumes the dialog.

If you use `pass` without a parameter, it passes everything the user enters until he or she tries to close the connection with the `quit` command. Then the dialog resumes, executes the remaining commands in the dialog, and closes the connection.

`pause` `pause`

Suspends capturing from within the dialog. If you turned `capture` on earlier in the dialog, you can pause capturing using the `pause` command.

To resume capturing later in the dialog, include a `capture` command (without a destination queue) at that point in the dialog.

`stop` `stop`

Stops capturing from within a dialog.

If you have turned `capture` on earlier in the dialog, you can turn it off using the `stop` command.

`timer` `timer <# of seconds> <string to display>`

Counts number of seconds specified in `<# of seconds>` when it is activated by the next `wait` command in the dialog.

When a specified string in the `wait` command is received, `timer` stops counting and `wait` resumes the dialog. If `wait` does not receive the expected string within the time specified in the `timer` command, `timer` displays the text specified in `string` and closes the connect session.

To use `timer`, follow it with the number of seconds you want it to count and the string you want it to display if that period of time elapses. For instance, you may want to use the `timer` command so it terminates the session if the dialog is unable to log in within 60 seconds.

When a `pass` command is active, an active `timer` command suspends counting. When `pass` command finishes, `timer` command resumes counting.

Also, the same `timer` command applies to any subsequent `wait` commands if no other `timer` commands appear before them. If you do not want to use the same `timer` value for another `wait` command later in the dialog, insert `timer 0` after the first `wait` command. This cancels the first `timer` command and causes subsequent `wait` commands to wait for their string forever if no other `timer` commands follow `timer 0`.

`type` `type <string to send>`

Sends a string to the device to which the service has connected.

For instance, if you were creating a dialog that types the user's name in response to a login prompt, you may use:

**`type joel smith`**

Most devices to which you connect expect a carriage return (represented by a `<cr>`) after each string you send. When this is the case, you must include a `<cr>` at the end of the string.

`wait`

`wait <string to wait for>`

Pauses the dialog until a specified string is received from a device to which the service has connected, or until a certain amount of time (specified by a `timer` command) has elapsed with no response.

To use this command, follow it with the string for which the dialog should wait. If you use `wait` without a parameter, the dialog waits until any character is received.

Unless a `timer` command has been executed first, the `wait` command waits forever until the specified string is received, so type exactly the string you want it to wait for, and keep in mind that `wait` is case-sensitive. If the dialog never receives the string `wait` is looking for, the dialog hangs, and you need to use the `quit connect` command to exit the dialog and return to your iNEWS Workstation.

# Appendix B

# System Files

This appendix contains samples of system files you are most likely to reference or change:

- /etc/dhcpd.conf
- /etc/hosts
- /site/config
- /site/printers/hplaser1
- /site/system
- /site/wires/anpa7
- console.cfg
- SYSTEM.CLIENT.WINDOWS
- SYSTEM.CONFIGURE.301-ACTION
- SYSTEM.MAP
- SYSTEM.RESOURCE
- SYSTEM.WIRES.DISTRIBUTION
- SYSTEM.WIRES.KEYWORDS
- SYSTEM.WIRES.KEYWORDS-AP
- SYSTEM.WIRES.KEYWORDS-AP2

# /etc/dhcpd.conf

```
allow bootp;
ignore unknown-clients;
not authoritative;
ddns-update-style none;
subnet 172.24.96.0 netmask 255.255.254.0{
```

PCU entry starts here...

```
host pcu10 {
    hardware ethernet 00:d0:b7:4b:52:f8;
    fixed-address 172.24.96.53;
}
```

MCSPC entry starts here...

```
host mcs19 {
    hardware ethernet 00:00:f8:20:25:19;
    fixed-address 172.24.96.71;
}
}
# line below because of dual nic cards
subnet 10.1.1.0 netmask 255.255.255.0 {}
```

# /etc/hosts

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost

# iNEWS Servers
123.123.123.95    nrcs-a1 a1 nrcs-a1.local
123.123.123.96    nrcs-b1 b1 nrcs-b1.local
10.1.1.1                  nrcs-a a nrcs-a.local
10.1.1.2                  nrcs-b b nrcs-b.local

# PCU
10.1.1.10         pcu10

# FTS
123.123.123.54    ftsserver
```

# /site/config

```
;
; iNEWS Starter config
;
host   ab a
        net     10
        reslist 121                     ; console connect
        servers 141 201 211             ; Print, Mail, Keyword
        servers 231 233 235 241 243 245 ; Seek & FTS
        servers 251 253 255 257 259     ; Action servers
        servers 271 273 275 277 279     ; Monitor servers
        servers 281 283 285 287 289     ; Monitor servers
        reslist 301 303 305 307 309     ; Txnet
        reslist 311 313 315 317 319     ; Rxnet
        reslist 321 323 325 327 329     ; Rxnet
        reslist 401:402                 ; Web Access
        reslist 501:504                 ; iNEWS sessions
;
host   ab b
        net     ; 20
        reslist 122                     ; net connect
        servers 232 234 236 242 244     ; Seek & FTS
        servers 252 254 256 258 260     ; Action servers
        servers 272 274 276 278 280     ; Monitor servers
        servers 282 284 286 288 290     ; Monitor servers
        reslist 302 304 306 308 310     ; Txnet
        reslist 312 314 316 318 320     ; Rxnet
        reslist 322 324 326 328 330     ; Rxnet
        reslist 601:604                 ; iNEWS sessions
;
host   a a
        net     10 ; 20
        reslist 121:122                 ; console, net connect
        servers 141 201 211             ; Print, Mail, Keyword
        servers 231:236 241:245         ; Seek & FTS
        servers 251:260                 ; Action servers
        servers 271:290                 ; Monitor servers
        reslist 301:310                 ; Txnet
        reslist 311:330                 ; Rxnet
        reslist 401:402                 ; Web Access
        reslist 501:504                 ; iNEWS sessions
        reslist 601:604                 ; iNEWS sessions
;
```

```
host   b b
        net     10 ; 20
        reslist 121:122                 ; console, net connect
        servers 141 201 211             ; Print, Mail, Keyword
        servers 231:236 241:245         ; Seek & FTS
        servers 251:260                 ; Action servers
        servers 271:290                 ; Monitor servers
        reslist 301:310                 ; Txnet
        reslist 311:330                 ; Rxnet
        reslist 401:402                 ; Web Access
        reslist 501:504                 ; iNEWS sessions
        reslist 601:604                 ; iNEWS sessions
;
; PCUS
;
pcu   10  pcu10  11:18
; pcu   20  pcu20  21:28
;
; PCU DEVICES
;
wire    11  9600  anpa7  AP  -  ;
wire    12  9600  anpa7  AX  -  ;
unused  13:18
;
; unused  21:28
;
; NETWORK RESOURCES
;
resource  121  console  -  -  ;
resource  122  net      -  -  ;
;
; SERVERS
;
server  141  printserver 141  -        ; slave printing
server  201  mailserver  201  -        ; mailserver
server  211  keyword     211  keyword  ; keyword server
;
; SEEK & FTS SERVERS
;
server  231  seek       231  -    ; seek server
server  232  seek       231  -    ; seek server
server  233  seek       231  -    ; seek server
server  234  seek       231  -    ; seek server
server  235  seek       231  -    ; seek server
```

```
server  236  seek       231  -    ; seek server
server  241  ftsindex   241  -    ; Fast Text Indexing
server  242  ftsseek    242  -    ; Fast Text Search
server  243  ftsseek    242  -    ; Fast Text Search
server  244  ftsseek    242  -    ; Fast Text Search
server  245  ftsseek    242  -    ; Fast Text Search
;
; ACTION SERVERS
;
server  251  action     251  -    ; timed-action
server  252  action     252  -    ; timed-action
server  253  action     253  -    ; timed-action
server  254  action     254  -    ; timed-action
server  255  action     255  -    ; timed-action
server  256  action     256  -    ; timed-action
server  257  action     257  -    ; timed-action
server  258  action     258  -    ; timed-action
server  259  action     259  -    ; timed-action
server  260  action     260  -    ; timed-action
;
; MONITOR SERVERS (Only 1 rundown per monitor server!)
;
server  271  monitor    271  -    ; Morning Show
server  272  monitor    272  -    ; 725 Cut-in
server  273  monitor    273  -    ; 755 Cut-in
server  274  monitor    274  -    ; 825 Cut-in
server  275  monitor    275  -    ; 855 Cut-in
server  276  monitor    276  -    ; 5PM
server  277  monitor    277  -    ; 6PM
server  278  monitor    278  -    ; 10PM
server  279  monitor    279  -    ; Updates-Afternoon
server  280  monitor    280  -    ; Updates-Evening
server  281  monitor    281  -    ;
server  282  monitor    282  -    ;
server  283  monitor    283  -    ;
server  284  monitor    284  -    ;
server  285  monitor    285  -    ;
server  286  monitor    286  -    ;
server  287  monitor    287  -    ;
server  288  monitor    288  -    ;
server  289  monitor    289  -    ;
server  290  monitor    290  -    ; Training
;
; RXNET/TXNET DEVICES
```

```
;
special  301  0  txnet  301  -    ; txnet
special  302  0  txnet  302  -    ; txnet
special  303  0  txnet  303  -    ; txnet
special  304  0  txnet  304  -    ; txnet
special  305  0  txnet  305  -    ; txnet
special  306  0  txnet  306  -    ; txnet
special  307  0  txnet  307  -    ; txnet
special  308  0  txnet  308  -    ; txnet
special  309  0  txnet  309  -    ; txnet
special  310  0  txnet  310  -    ; txnet
;
special  311  0  rxnet    -  -    ; rxnet
special  312  0  rxnet    -  -    ; rxnet
special  313  0  rxnet    -  -    ; rxnet
special  314  0  rxnet    -  -    ; rxnet
special  315  0  rxnet    -  -    ; rxnet
special  316  0  rxnet    -  -    ; rxnet
special  317  0  rxnet    -  -    ; rxnet
special  318  0  rxnet    -  -    ; rxnet
special  319  0  rxnet    -  -    ; rxnet
special  320  0  rxnet    -  -    ; rxnet
special  321  0  rxnet    -  -    ; rxnet
special  322  0  rxnet    -  -    ; rxnet
special  323  0  rxnet    -  -    ; rxnet
special  324  0  rxnet    -  -    ; rxnet
special  325  0  rxnet    -  -    ; rxnet
special  326  0  rxnet    -  -    ; rxnet
special  327  0  rxnet    -  -    ; rxnet
special  328  0  rxnet    -  -    ; rxnet
special  329  0  rxnet    -  -    ; rxnet
special  330  0  rxnet    -  -    ; rxnet
;
; WEB ACCESS SESSIONS
;
websession  401:402
;
; INWS SESSIONS
;
inws  501:504  -  0  gnews  -     ; inws
inws  601:604  -  0  gnews  -     ; inws
;
; END OF FILE
```

# /site/printers/hplaser1

```
; HP LaserJet IVsi Printer profile
;
ejectcode      <ff>
ejectcount     1
idlecount      0
pagelength     66
scriptrhstart  32
scriptrhmax    23
scriptlhmax    25
scriptshift    yes
scripttemplate no
;
expand  <esc>(0U<esc>(sp7.0h18vs0b11T  <nul>   ;Courier
;
font 2  <esc>(s3B           <esc>(s0B          ;Bold
font 3  <esc>(s4B           <esc>(s0B          ;extra bold
font 4  <esc>&dD            <esc>&d@           ;underline
font 5  <esc>&dD<esc>(s3B   <esc>&d@<esc>(s0B  ;bold/Under
;
;form 7  <esc>&l0O<esc>(s10H<esc>(s12V <nul> ;landscape,
;form 8  <esc>&l1H      <nul>               ;top tray
;form 9  <esc>&l4H      <nul>               ;lower tray
```

# /site/system

```
id=NRCS       net=AB
lowwater=5000  highwater=6250     purgelimit=5
load=5         pausetimeout=0:05
;
; defaults - if parameter not present these values are assumed
;            unless overridden on the command line.
;
; auto_upgrade=yes     lowwater=1250          remotetimeout=0:00
; clockmax=12          maxhits=500            scriptlhmax=40
; excluded_video=none  min_passwd_length=5    scriptrhmax=40
; highwater=2500       msgserver=silent       security=or
; lastlogin=yes        pausetimeout=0:30      timechar=:
; load=0               purgelimit=0           timer=verbose
; localtimeout=0:00    readrate=180           wordlength=0
```

# /site/wires/anpa7

```
; %Z% %M%     %I%           Date %E%      %Q%
; anpa7 - suitable for upi7, ap7, and cupi7

bits      7
start     <soh>
eos                 <EOT>
category  -------
slug      30
;
;
map       <16>      <SP>          ; TTS SPACE BAND
map       <25>      <SP>          ; EM SPACE
map       <29>      <SP>          ; THIN SPACE
map       <30>      <SP>          ; EN SPACE
map       <35>      .             ; EN LEADER
map       <42>      ..            ; EM LEADER
map       <60>      <23>          ; HARDEOL
map       <61>      <23>          ; HARDEOL
map       <62>      <23>          ; HARDEOL
map       <64>      |N
map       <91>      1/8
map       <92>      1/4
map       <93>      3/8
map       <94>      |B
map       <95>      --
map       <123>     1/2
map       <124>     5/8
map       <125>     3/4
map       <126>     7/8
```

# console.cfg

```
log b:log 32764
computer                ; A server
name            a
label           NRCS
irq             3
hostess         2c0
speed           9600
;
computer                ; B server
name            b
label           NRCS
irq             3
hostess         2c8
speed           9600
;
computer                ; C server
name            c
label           NRCS
irq             3
hostess         2d0
speed           9600
;
modem                   ; remote console
password        issecret
timeout         6:00
irq             3
hostess         2d8
speed                   2400
```

# SYSTEM.CLIENT.WINDOWS

*Either IP address or network card information is acceptable; IP address is preferable.*

```
125.1.100.1  ;02608cdbe7a2 ns001 big table
125.1.100.2  ;02608cd95e7e ns002 brock 07
125.1.100.3  ;02608c7e178e ns003 nydam 16
125.1.100.4  ;02608c7e67aa ns004 lockett 38
02608c7e519f ;ns005 michel 87
02608c7e1790 ;ns006 thibault 22
02608c7e51a8 ;ns007 ries   04
02608c7e6c01 ;ns008 lucas 57
125.1.100.9  ;02608c7e52c6 ns009 christensen 48
125.1.100.10 ;02608c7e5260 ns010 betty 69
125.1.100.11 ;0020aff431ff ns011 aveson 28
125.1.100.12 ;02608c7e6bfc ns012 robinson 49
125.1.100.13 ;02608c7e5274 ns013 tinsley 63
125.1.100.14 ;02608c7e6b58 ns014 reed 44
125.1.100.15 ;02608c7e532e ns015 landis 61
125.1.100.16 ;0020aff42efc ns016 dorr 51
125.1.100.17 ;0020aff42dad ns017 donnallen 60
125.1.100.18 ;0020aff42dcb ns018 douda 54
125.1.100.19 ;0020aff432ca ns019 kennedy 52
125.1.100.20 ;0020aff42d42 ns020 control room
125.1.100.21 ;0020aff431bf ns021 becker 30
125.1.100.22 ;0020aff43277 ns022 glass 29
```

# SYSTEM.CONFIGURE.301-ACTION

```
scan      system.shredder
     remove

scan      system.cables.master
     dup  system.cables.groups
     dup  system.cables.cable#
     dup  system.cables.device_type

scan      phones.airports
     dup  phones.*all
scan      phones.business
     dup  phones.*all
scan      phones.fire
     dup  phones.*all
scan      phones.government.federal
     dup  phones.*all
scan      phones.government.local
     dup  phones.*all
scan      phones.government.state
     dup  phones.*all
scan      phones.hospitals
     dup  phones.*all
scan      phones.media
     dup  phones.*all
scan      phones.misc
     dup  phones.*all
```

# SYSTEM.MAP

```
;
;<rundown queue>    <event dir>    <composite><grp><mon off time>
;<name>   <dev name>     <update>  <mct temp> <drv><dir><range>
;
show.6am.rundown     show.6am                -       monitor
800
cg        cg@casvr             update     ca-cg     C:AMNEWS:500
599
cart      cart@casvr2           update     ca-cart   3

show.cutins.725      show.cutins             -       monitor
900
cg        cg@casvr             update     ca-cg     C:AMNEWS:700
750
cart      cart@casvr2           update     ca-cart   3

show.noon.rundown    show.noon               -       monitor
1400
cg        cg@casvr             update     ca-cg     C:AMNEWS:800
899
cart      cart@casvr2           update     ca-cart   3

show.6pm.rundown     show.6pm                -       monitor
2100
cg        cg@casvr             update     ca-cg     C:PMNEWS:500
599
cart      cart@casvr2           update     ca-cart   3

show.11pm.rundown    show.11pm               -       monitor
0300
cg        cg@casvr             update     ca-cg     C:PMNEWS:500
599
cart      cart@casvr2           update     ca-cart   3

show.updates.rundown show.updates            -       monitor
0100
cg        cg@casvr             update     ca-cg     C:PMNEWS:800
850
cart      cart@casvr2           update     ca-cart   3

show.special.rundown show.special            -       monitor
0200
cg       cg@casvr             update     ca-cg     C:PMNEWS:4000
```

```
4100
cart      cart@casvr2           update     ca-cart   3

show.sat-am.rundown  show.sat-am             -       monitor
1100
cg        cg@casvr              update     ca-cg     C:PMNEWS:500
599
cart      cart@casvr2           update     ca-cart   3

show.sports-sat.rundown show.sports-sat      -       monitor
2200
cg        cg@casvr              update     ca-cg     C:PMNEWS:800
899
cart      cart@casvr2           update     ca-cart   3

show.sat-am.cut-in-1 show.sat-am             -       monitor
1100
cg        cg@casvr              update     ca-cg     C:AMNEWS:800
850
cart      cart@casvr2           update     ca-cart   3
```

# SYSTEM.RESOURCE

```
;Dev Style    Template  Effect     # lines   Comment
cg   bnews      62        -          0         ;breaking news
cg   th         76        -          1         ;today in history
cg   mm         75        -          1         ;money matters
cg   pball      9996      -          6         ;powerball
cg   lottopb    9993      -          12        ;lotto/powerball
cg   cashpb     9994      -          9         ;cash3/powerball
cg   world      50        -          1         ;the world
cg   nation     50        -          1         ;the nation
cg   fancash    9995      -          8         ;fantasy 5 cash 3
cg   south      50        -          1         ;the south
cg   recap      52        -          1         ;recap
cg   nitetz     53        -          1         ;11 tonight
cg   tvex       67        -          0         ;tv exclusive
cg   uv         68        -          0         ;unedited video
cg   tji        69        -          0         ;this just in
cg   intv1      1         -          1         ;1 line interview
cg   intv       2         -          2         ;2 line interview
cg   intvdate   2         -          2         ;1 line interview
with date
cg   loc        4         -          1         ;1 line locator
cg   locdate    5         -          2         ;locator with date
cg   date       6         -          1         ;date super
cg   file       7         -          0         ;file tape
cg   newsfile   7         -          0         ;newsfile
cg   ctsy       8         -          1         ;courtesy
cg   sktch      10        -          1         ;sketches by
cg   spkng1     11        -          1         ;speaking one line
cg   spkng      12        -          2         ;speaking two line
cg   scenes     13        -          1         ;scenes from
cg   locskylv   24        -          1         ;live skycam locator
cg   locsky     16        -          1         ;locator skycam
cg   intvskylv  17        -          2         ;interview skycam live
cg   repsky     18        -          1         ;reporter skycam
cg   loclv      19        -          1         ;live locator
cg   intvlv     20        -          2         ;live interview
cg   intv1lv    28        -          1         ;live interview 1 line
cg   live       21        -          0         ;live
cg   replv      22        -          1         ;live generic reporter
cg   lvsky      23        -          0         ;live skycam
cg   rep        27        -          1         ;reporter not live
cg   sky        141       -          0         ;skycam live
```

```
cg   frsky     254       -         0         ;from skycam
;
;special for NoonDay
;
cg   locn      1002      -         1         ;1-line
locator/noonday
cg   scenesn   1022      -         1         ;scenes from/noonday
cg   innd      1001      -         2         ;2 line interview noon
cg   lond      1008      -         1         ;live locator noon
cg   inlvnd    1007      -         2         ;live 2 line interview
noon
;
; SPORTS
cg   nl        950       -         10        ;national league
cg   al        951       -         10        ;american league
cg   nfl       952       -         10        ;nfl
cg   cf        953       -         10        ;college football
cg   nba       954       -         10        ;nba
cg   cbb       955       -         10        ;college basketball
cg   nhl       956       -         10        ;nhl
cg   prep      957       -         16        ;high school
cg   scr       960       -         4         ;score lower third
;; Anchors/Reporters

cg   sa        104       -         0        ; steve aveson
cg   salv      105       -         0        ; steve aveson live
cg   sasky     107       -         0        ; steve aveson skycam
```

# SYSTEM.WIRES.DISTRIBUTION

```
AP#medc2##          wires.medsource
AP#env####          wires.environmental
AP#######a#         wires.national
AP#######c#         wires.features
AP#######d#         wires.summaries
AP#######e#         wires.features
AP#######f#         wires.business
AP#######g#         wires.state/regional
AP#######h#         wires.summaries
AP#######i#         wires.international
AP#######j#         wires.state/regional
AP#######m#         wires.farm
AP#######n#         wires.state/regional
AP#######o#         wires.weather
AP#######p#         wires.politics
AP#######q#         wires.sports.all
AP#######q#         wires.sports.scores
AP#######r#         wires.advisory.other
AP#######s#         wires.sports.all
AP#######s#         wires.sports.stories
AP#nbc##t#          wires.newschannel
AP#######t#         wires.advisory.other
AP#######u#         wires.state/regional
AP#######v#         wires.advisory.other
AP#######v#         wires.daybook
AP#######w#         wires.national
AP########          wires.unknown
AP########          wires.ap                ALWAYS
AP########          wires.all               ALWAYS    URGENT
NC#######u#         wires.advisory.priority
NC########          wires.all
NC########          wires.newschannel
NC########          wires.all               ALWAYS    URGENT
AP#######q#         sports.wires
AP#######s#         sports.wires
```

# SYSTEM.WIRES.KEYWORDS

```
* AP
system.wires.keywords-ap          wires.keywords
system.wires.keywords-ap2         wires.
* UP
*
system.wires.keywords-others      wires.keywords
```

# SYSTEM.WIRES.KEYWORDS-AP

```
destination        atlanta
atlanta
destination        rape
rape
destination        braves
braves
destination        carter
carter
destination        tuscaloosa
tuscaloosa and football
destination        xgr
xgr
destination        medsource
medc2 or medstar
destination        gingrich
gingrich
destination         space
(space AND shuttle) OR (shuttle AND columbia) or (shuttle AND
atlantis)
shuttle AND discovery
destination         drugs
crack AND cocaine
destination         air-safety
crash and plane OR ntsc or n.t.s.c. or faa or f.a.a.
destination         delta
(delta and airlines) OR (hartsfield and airport)
destination        guns
guns or weapons or nra
```

# SYSTEM.WIRES.KEYWORDS-AP2

```
destination 9am-specials
900a and specials
```

# Appendix C

# Standard Dictionaries

This appendix defines and explains contents of standard dictionaries as they are installed on your iNEWS newsroom computer system. Reference this information when modifying dictionary contents. Major sections are:

- Using Dictionaries to Define Messages and Commands
- Customizing Dictionaries
- Utility Messages Dictionary (/site/dict/messages)
- ccumsgs Dictionary (/site/dict/ccumsgs)
- Commands Dictionary (/site/dict/ccucmds)
- Queues Dictionary (/site/dict/queues)
- Words Dictionary (/site/dict/words)
- Connect Dictionary (/site/dict/doac)
- Telex Dictionary (/site/dict/telex)
- Dial Dictionary (/site/dict/dial)
- Keyboard Macros Dictionary (/site/dict/keymacros)
- Printer Messages Dictionary (/site/dict/printmsgs)
- Case-shifting Dictionary (/site/dict/shift)
- MCS Dictionary (/site/dict/mcs)

# Using Dictionaries to Define Messages and Commands

Most commands, messages, and many queues your iNEWS newsroom computer system uses are defined in dictionaries. Your system has a number of these dictionaries, each defining a particular group of commands, messages, or words, such as: the names of all commands are defined in the `ccucmds` dictionary. Many messages your system uses are defined in the `ccumsgs` dictionary.

Dictionaries let you customize your system's messages, workstation and commands, as well as the names of many of the queues your system uses (such as `SYSTEM.SEEK`). You can change the names of any of your system's required queues by editing their definitions in the `queues` dictionary and then having your system read the modified dictionary.

*If you customize your dictionary entries, keep a log of the changes you make so your changes can be re-entered after future software upgrades.*

The following table lists names and locations of dictionary files.

**Names and Locations of Dictionary Files**

| File Name & Location | Dictionary File Name |
|---|---|
| `/site/dict/ccucmds` | Terminal commands |
| `/site/doct/ccumsgs` | Error and Status Messages |
| `/site/dict/ccuvideo` | Video Modes |
| `/site/dict/messages` | Utility Messages |
| `/site/dict/queues` | Required Queues |
| `/site/dict/words` | Status and Option Words |
| `/site/dict/doac` | Connect Commands and Messages |
| `/site/dict/telex` | Telex Commands and Messages |
| `/site/dict/dial` | Modem/Dial Messages |

**Names and Locations of Dictionary Files**

| File Name & Location | Dictionary File Name |
|---|---|
| /site/dict/keymacros | Key Names for Macros |
| /site/dict/printmsgs | Printer Messages |
| /site/dict/shift | Case-Shifting Parameters |
| /site/dict/mcs | MCS Alarms and Words |
| /site/dict/mctcmds | MCT Commands |
| /site/dict/mctmsgs | MCT Error and Status Messages |

Each line in a dictionary begins with a standard name followed by the translation for the command, message, or word represented by that standard name, as shown in the following Dictionary Standard Names and Translation table..

**Dictionary Standard Names and Translation**

| Standard Name | Translation |
|---|---|
| cursor left | /cu l |
| cursor top | /cu t |
| cursor up | /cu u |

Each translation is preceded by a slash (/). In commands, the translation represents the minimum a user must enter to execute the command. In the example above, the translation /cu u is the minimum a user must enter to execute the cursor up command.

The translations may be more than one word long, but the dictionaries have limited space, so keep each translation as short as possible. As a rule, make translations just long enough to be unique from any other translation. For instance, the cursor up command is translated as cu u. This is the shortest translation for this command that keeps the translation unique.

If the purpose of a standard name or its translation is not clear, clarify them by inserting a comment on the line following the translation. Begin the comment line with a semicolon ( ; ) to prevent the system from treating it as part of the translation.

*iNEWS newsroom computer system uses standard names in each dictionary to match each translation to the correct command, message, or word. Do not change any of the standard names in your dictionaries.*

# Customizing Dictionaries

Your system's dictionaries are text files stored in the `/site/dict` directory. Because they are text files, you can change any dictionary translation using `ed`, the line editor, at the console. See Appendix H for more information.

## Changing Default Dictionary Values

As an example of how to modify a dictionary translation, change the enter directory and enter queue commands to make directory and make queue. To do this, change the translations for enter directory and enter queue from `e d` and `e q` to `ma d` and `ma q`, respectively.

The new translations, like the ones they are replacing, are as short as they can be and still remain unique. Keep each translation short, because there is a space limit.

If necessary, hide commands from users by changing the command name. For instance, to restrict access to the `order` command, change it to something like `arrange` and inform only users that you want to have access to the command.

**To edit the dictionary file, do the following:**

1. Select *all* servers at the console, so changes you make are made to each server's copy of the file.

   The `enter directory` and `enter queue` commands have their translations stored in `/site/dict/ccucmds`.

This procedure uses the `ed` line editor. See Appendix H for more information.

2. Open /site/dict/ccucmds for editing by typing:
   **`ed /site/dict/ccucmds`**

   A message similar to the following appears:

```
editing /site/dict/ccucmds
1824
```

3. Move to the line containing the standard name `enter` directory by typing:
   **`enter directory /e d`**
4. Change `e d` to `ma d` by typing **`s/e d/ma d`**.
   **`enter directory /ma d`**
5. Change the translation for `enter queue` in the same way.
   **`enter queue /ma q`**
6. Type **`w`** (lowercase w) to save your changes to disk and type **`q`** to quit edit.
   **`w`**
   `1826`
   **`q`**

**To change a default dictionary value, do the following:**

1. Edit the dictionary file.

   See the previous procedure.
2. Type a message like the following:

```
NRCS-A: broadcast System going down at 1:55 pm. Please log
out.
```

3. Type **`offline`** to bring the system offline. This will prevent users from logging in.
4. At the designated time, type **`list t`** to see whether anyone is still logged in.

   If there is someone logged in, a message similar to the following appears:

```
G500 terryl      A
```

5. If anyone is logged in, type **`logout all`** to log them out.
6. Type **`stop all`** to stop all servers.
7. Become a console superuser.
8. Type **`makeccutab -b`** (-b for build tables).

   Use `makeccutab` to build the command and error message tables in the PCU program.

   Information similar to the following appears::

```
Translating </site/dict/ccucmds>
Translating </site/dict/ccuvideo>
Translating </site/dict/ccumsgs> and </site/dict/queues>
<atccu> unused space is 205 characters
<directccu> unused space is 205 characters
```

   After modifying a dictionary, run `makeccutab` and `maketab` to have your system read the modified dictionaries and incorporate changes into its programs.

   The `makeccutab` command builds the command and error message table for the PCU dictionaries and then displays how much space is unused in these dictionaries. In the previous example, the unused space is 205 characters.

9. Type `maketab -b` to build command and message tables and translate dictionaries for the iNEWS newsroom computer system.

   When the prompt returns, run `maketab` to translate site dictionaries. To make it build the table, add `-b` to this command.

   A display similar to the following appears:

```
Translating </site/dict/queues>
Translating </site/dict/words>
Translating </site/dict/messages>
<host> unused space is 109 characters
Translating </site/dict/doac>
<snews> unused space is 221 characters
Translating </site/dict/doac>
<nxserver> unused space is 221 characters
```

   The `maketab` command translates each dictionary and then displays unused space in these dictionaries.

If you changed the name of a command in a command dictionary, you must also change the function key definition that references that command.

10. Restart all devices.
11. At the prompt, exit from superuser using Ctrl+D.
12. Reset your system's PCUs using their reset switches.
13. When all PCUs have completed their reset, restart them by typing restart all at the console.

    Once you have restarted the system, try using **`enter directory`** and **`enter queue`**. The system no longer recognizes these commands. It recognizes the **`make directory`** and **`make queue`** commands.

14. Back up your site files with the **`sitedump`** command.

    If you do not have one of your site files, a message similar to the following appears when you run the **`makeccutab`** or **`maketab`** console command:

```
Translating </site/dict/ccucmds>
Cannot access translation file </site/dict/ccucmds>.
Do you wish to use the standard English translations and
continue? (n/y)
```

If a set of dictionaries exceeds the amount of space allotted, a message similar to the following appears:

```
Table space exceeded by 14 characters
No modifications done!
```

## Restoring Dictionary Defaults

You can restore original dictionary default translations without editing the dictionary again. Original dictionary files are stored in the `/tmp/dict` directory and have a pound symbol (#) prefix, such as `#mcs`. So, even after you have made changes to a dictionary, you can restore the standard default translations by copying the appropriate dictionary file from `/site/dict` and running the **`makeccutab`** and **`maketab`** commands again.

### To restore standard translations for `enter directory` and `enter queue`, do the following

1. List files in `/site/dict` by running the following `ls` command:

   ls /site/dict

   A display similar to the following appears:

```
#ccucmds          #pcdgidriver      ccumsgs
pcdgidriver
#ccumsgs          #pcflexidriver    ccuvideo
pcflexidriver
#ccuvideo         #pcinfdriver      convertvideo
pcinfdriver
#dial             #pclsfdriver      dial
pclsfdriver
#doac             #pcqcbdriver      doac
pcqcbdriver
#keymacros        #pcqdeltradriver  keymacros
pcqdeltadriver
#mcs              #pcqpbdriver      mcs
pcqpbdriver
#mctcmds          #pcsvsdriver      mctcmds
pcsvsdriver
#mctmsgs          #printmsgs        mctmsgs         printmsgs
#messages         #queues           messages        queues
#pca72driver      #shift            pca72driver     shift
#pcavidapdriver   #telex            pcavidapdriver  telex
#pcbcartdriver    #words            pcbcartdriver   words
#pccypherdriver   ccucmds           pccypherdriver
```

2. Type `cp` followed by the pathname of the file you want to copy, and the pathname of the file you want to contain the copy. For instance,

   **`cp /tmp/dict/ccucmds /site/dict/ccucmds`**

   After you copy the file to /site/dict, complete the procedure for changing a translation.

# Utility Messages Dictionary (/site/dict/messages)

The `/site/dict/messages` dictionary holds a number of utility messages displayed in utility programs used by the iNEWS newsroom computer system including messages usually displayed when a user is building a form or creating a keyboard story. A few console messages are also included in this dictionary.

These messages do not contribute to the total size of translations, because they are sent to workstations only under special circumstances, and—with some exceptions—the system looks up translations only as they are needed rather than building them into a program. Some exceptions include the following messages, which are built into programs via the `maketab` program:

- M_COMPUTER
- M_KEYBAD
- M_LASTLOG
- M_NOBULKPR
- M_ONDEVICE
- M_PIOFAIL
- M_WIREFAIL
- M_WIREIDLE
- M_WIRERESUME

*Some messages defined in dictionary entries on the server that would typically be displayed in the iNEWS client status bar can now have a display method designator to determine if the message should be displayed in the status bar or as a pop-up message. This display method designator is based on the first character of the message definition. A display designator of 1 denotes that the message will appear on the status bar. A display designator of 2 denotes the message displays as a pop-up dialog in addition to being displayed in the status bar and alerts history. The default display method will be a '1' if no display designator is specified. Pop-up dialogs do not steal focus away from the user or interrupt their work.*

This section describes these utility messages in the following tables:

- DBServer Program Messages
- Disconnect Program Messages
- Category and Keyword Check Program Messages
- Keyboard Check Program Messages
- Keyboard Check Program Messages for Macros
- Grpcheck Messages
- Wire Program Messages
- Mail Server Messages
- Validation (Action) Server
- Seek Server Messages
- Last Login Messages
- Messages for Print Server
- Message for the Sony Barcode Printer
- dbtraits Messages

For information on the display method designator appearing in the message definition, see note on page 203.

### DBServer Program Messages

| Standard Name | Translation |
|---|---|
| M_NOSPACE | /2NO SPACE IN SYSTEM |
| M_LOWSPACE | /1SYSTEM LOW ON SPACE |

For information on the display method designator appearing in the message definition, see note on page 203.

### Disconnect Program Messages

| Standard Name | Translation |
|---|---|
| M_DISCONNECT | /2COMPUTER DISCONNECTED |

### Category and Keyword Check Program Messages

| Standard Name | Translation |
|---|---|
| M_NODEST | /No destination found |
| M_DUPEDEST | /Duplicate destination |
| M_LINE | /Line |
| M_KWFMAX | /Too many keyword distribution files |
| M_BADQUEUE | /Could not enter the queue |
| M_NOTQUEUE | /Not a queue |
| M_PURGEZERO | /Queue is never purged |
| M_SYSERROR | /System error |
| M_CATLONG | /Category code word too long |
| M_CATMAX | /Too many category codes |
| M_CATFORM | /Illegal category format |
| M_CATHIDE | /Hidden category |
| M_KWDLONG | /Keyword too long |
| M_KWDMAX | /Too many keywords |
| M_WNOTLAST | /Default keyword list must be last |
| M_SYNERROR | /Syntax error |

| Standard Name | Translation |
|---|---|
| M_MISSING | /Missing |
| M_UNEXPCTD | /Unexpected |
| M_FILENUM | /Maximum file number bad |
| M_UPDATEQ | /UPDATE queue |
| M_INVALIDQ | /Invalid destination queue |

### Keyboard Check Program Messages

| Standard Name | Translation |
|---|---|
| M_KEYDUP | /Duplicate key description |
| M_KEYRANGE | /Invalid key number |
| M_KEYSEP | /Missing key number separator (~) |
| M_KEYSTART | /First key description does not begin with @ |
| M_KEYMIN | /Not enough key descriptions |
| M_KEYLONG | /Keyboard description contains too many characters |
| M_KEYREP | /Warning: a key definition contains a repeating function |
| M_KEYFUNKY | /Warning: badly placed @ exists in key definition line |
| M_KEYOK | /Keyboard ok |
| M_KEYBAD | /Keyboard NOT usable |
| M_COMPUTER | /Computer |

### Keyboard Check Program Messages for Macros

| Standard Name | Translation |
|---|---|
| M_MACRO | /%s macro #%d: |
| M_NOLOCATE | /could not locate "%c%d" |
| M_BADMEMORY | /memory allocation error |
| M_REFERENCE | /circular reference to macro #%d: |
| M_BADSTACK | /unable to stack keywords |
| M_MISMATCH | /mismatched "%c%c" |
| M_TWOTILDES | /Multiple "%c"s found |
| M_RESWORD | /no "%c%c" found for reserved word %s |
| M_TWOTAGS | /multiple macro keys: %s %s |
| M_NOTILDE | /no "%c" found |
| M_NOTAG | /no macro key tag |
| M_UNKNOWN | /unknown macro key identifier: %s |
| M_IGNORING | /ignoring: %s |
| M_UNEXPECTED | /unexpected: %s |
| M_TWODEFS | /duplicate macro definition: |
| M_LONESTATE | /isolated keyboard state: |
| M_DISTRIBUTE | /%s does not distribute |
| M_EMPTY | /empty macro |
| M_NUMKEYNOSHIFT | /Shifted numeric keypad 0 - 9 keys cannot be assigned macros |

| Standard Name | Translation |
|---|---|
| M_STDHELP | /Warning: "Help" key redefined: |
| M_STDCELLEDIT | /Warning: "Edit Cell" key redefined: |
| M_STDFINDNEXT | /Warning: "Find Next" key redefined: |
| M_STDEXIT | /Warning: "Exit" key redefined: |
| M_STDCLOSE | /Warning: "Window Close" key redefined: |
| M_STDDISCARD | /Warning: "Discard Changes" key redefined: |
| M_STDREFRESH | /Warning: "Refresh" key redefined: |
| M_STDTOGGLETEXTFORM | /Warning: "Toggle Story Form" key redefined: |
| M_STDSCRIPT | /Warning: "Script Swap" key redefined: |
| M_STDPRIORITYQUEUE | /Warning: "GoTo Priority Queue" key redefined: |
| M_STDALERTSHISTORY | /Warning: "GoTo Alerts History" key redefined: |
| M_STDMESSAGEBAR | /Warning: "Communicate Message Bar" key redefined: |
| M_STDMESSAGEHISTORY | /Warning: "Communicate Message Show History" key redefined: |
| M_STDMAIL | /Warning: "Communicate Open/Close Mail" key redefined: |
| M_BADPAUSE | /Invalid pause interval |

### Grpcheck Messages

| Standard Name | Translation |
|---|---|
| M_GRPBADALIAS | /Invalid name follows word "alias" |
| M_GRPBADGRP | /Invalid name follows word "group" |
| M_GRPBADDEV | /Not a workstation device |
| M_GRPERRSKIP | /Group or alias word missing. Skipping text |
| M_GRPEXINPUT | /Ignoring words following group name |
| M_GRPAEXINPUT | /Ignoring words following alias name |
| M_GRPINT | /Internal groupchecker error |
| M_GRPDBERR | /Failed to access first story |
| M_GRPDEVSYN | /Bad workstation device specification |
| M_GRPNAMUSR | /User name used as group or alias name |
| M_GRPNAMDUP | /Duplicate group or alias name |
| M_GRPNONAME | /Missing group name |
| M_GRPALNONAME | /Missing alias name |
| M_GRPGRPREC | /Recursive group membership |
| M_GRPINALIAS | /Name already used as group name |
| M_GRPAINGRP | /Name already used as alias name |
| M_GRPMEMBAD | /Not a user or workstation |
| M_GRPNIX | /No groups or aliases found |
| M_GRPRESERV | /Improper use of reserved word |

| Standard Name | Translation |
|---|---|
| M_GRPNOSTORY | /Failed to open story |
| M_GRPNOQ | /Failed to open queue |
| M_GRPFLODEF | /Cannot open default aliases file |
| M_GRPFLONEW | /Cannot open new aliases file |
| M_GRPFLSOLD | /Cannot save old aliases file |
| M_GRPFLLONG | /more than 50,000 alias names created |
| M_GRPOK | /GROUPS story OK |
| M_GRPBAD | /GROUPS story NOT OK |
| M_GRPSOSO | /GROUPS story accepted, with errors |
| M_GRPNOSOUSER | /Lookup of "so" user in passwd file failed. |
| M_GRPCHOWNFAILED | /Could not change ownership of mail alias file. |

### Wire Program Messages

| Standard Name | Translation |
|---|---|
| M_WIREFAIL | /1HOST-CCU COMMUNICATION ERROR |
| M_WIREIDLE | /1wire has been idle for |
| M_WIRERESUME | /1wire received story, was idle for |

### Mail Server Messages

| Standard Name | Translation |
|---|---|
| M_MAILSYNTAX | /Can't send mail, no address |
| M_MAILNOREC | /Unable to receive mail from |
| M_MAILQUEUE | /Can't return mail, bad mail queue |

### Validation (Action) Server

| Standard Name | Translation |
|---|---|
| M_VALID | /Story valid |
| M_INVALID | /Story invalid |
| M_VMOVEDTO | /Story invalid - Moved to |

Seek Server Messages are also used by `ftsseek`.

### Seek Server Messages

| Standard Name | Translation |
|---|---|
| M_BGSNSCHP | /No search path |
| M_BGSNRESP | /No results path |
| M_BGSSCHTP | /Invalid Search Type |
| M_BGSIKWEX | /Invalid keyword expression |
| M_BGSSPDNE | /Search Path does not exist |
| M_BGSRQDNE | /Results Queue does not exist |

| Standard Name | Translation |
|---|---|
| M_BGSRQNAQ | /Results Queue not a queue |
| M_BGSRQNPM | /No write permission for results queue |
| M_BGSINVLD | /Invalid Search |
| M_BGSEOP | /end of path |
| M_BGSMAXH | /max hits |
| M_BGSSPI | /Invalid search path |
| M_BGSRQI | /Invalid results queue |
| M_BGSRQOE | /Open error on results queue |
| M_BGSDONE | /Done |
| M_BGSSTOPPED | /Stopped |
| M_BGSCANCELLED | /Cancelled |
| M_BGSSYNTAXERROR | /Syntax Error |
| M_BGSCOMMERROR | /Communication Error |
| M_BGSMPAT | /Missing search pattern |

### Last Login Messages

| Standard Name | Translation |
|---|---|
| M_LASTLOG | /Last Login |
| M_ONDEVICE | /on device |

### Messages for Print Server

| Standard Name | Translation |
|---|---|
| M_PRINTER_BUSY | /Printer Is OFFLINE |

### Message for the Sony Barcode Printer

| Standard Name | Translation |
|---|---|
| M_NOBULKPR | /Bulk printing disabled |

### dbtraits Messages

| Standard Name | Translation |
|---|---|
| M_CHANGE_GROUP | /1Group, form, order, or sortfield changed. |

# ccumsgs Dictionary (/site/dict/ccumsgs)

The following table shows entries used for PCUs in the ccumsgs dictionary, with their default translations.

### PCU Messages

| Standard Name | Translation |
|---|---|
| D_AIRBEND | /End-backwards |
| D_AIRFEND | /End-forward |
| D_AIRSTOP | /Stopped |
| D_AUTHORIZED | /Not allowed |
| D_BACKT | /Backtime |
| D_BADARG | /Bad argument |
| D_BADDEST | /Bad destination |
| D_BADKBD | /Bad keyboard |
| D_BADREP | /Must FIND First |
| D_BUSY | /Busy |
| D_CANTFIND | /Can't find |
| D_CMD | /CMD |
| D_CONTINUE | /Press CMD to continue |
| D_DEFINED | /Text is defined |
| D_DIRMODE | /Not in directory |
| D_DUPED | /2 spaces then DUP'D |
| D_EMPTY | /5 spaces then EMPTY |
| D_ERROR | /System error |
| D_EXISTS | /Already exists |
| D_GONE | /Gone |
| D_HOME | /home |
| D_INSERT | /Insert |

| Standard Name | Translation |
|---|---|
| D_KEY | /Key (must match ccucmd translation) |
| D_KILLED | /1 space then KILLED |
| D_LINE | /Line |
| D_LOCKED | /Locked |
| D_LOGGEDIN | /Logged in |
| D_LOGIN | /Login (must match ccucmd translation) |
| D_MAIL | /mail |
| D_MAILSTORY | /Story |
| D_MAXSONS | /Too many |
| D_MESSAGE | /Message |
| D_MODE | /Invalid in this mode |
| D_MOVED | /2 spaces then MOVED |
| D_NAMETL | /Name too long |
| D_NEW_PASSWORD | /New password |
| D_NOARG | /Needs argument |
| D_NOCMD | /Unrecognized command |
| D_NODEFINED | /Nothing defined |
| D_NODELETED | /Nothing deleted |
| D_NOFINDSTR | /Type word to find |
| D_NOLOG | /Not logged in |
| D_NOMSG | /No messages |
| D_NOREPSTR | /Type new word |
| D_NOTBATCH | /Queue lacks that capability |
| D_NOTEMPTY | /Not empty |

| Standard Name | Translation |
|---|---|
| D_NOTEXIST | /Story moved |
| D_NOUSER | /No such user |
| D_OFFLINE | /Offline |
| D_ONAIR | /On air |
| D_ORDER | /Order |
| D_PASSWORD | /Password (must match ccucmd translation) |
| D_PAUSED | /Paused |
| D_PERM | /P |
| D_PRINTED | /PRINTED |
| D_PURGELOCK | /Queue busy |
| D_QONLY | /Only in queue |
| D_QUEMODE | /Not in queue |
| D_REFRESH | /Refresh |
| D_REPLACE | /Replace |
| D_RONLY | /Read-only |
| D_SORRY | /Sorry |
| D_SORTED | /Sorted |
| D_STORY | /Story |
| D_SYNTAX | /Bad usage |
| D_TOO_SHORT | /Too short |
| D_TXED | /TRANSFERRED |
| D_UNABLEMAIL | /Unable to send mail |
| D_UNKNOWN | /Unknown command |
| D_WAIT | /Please stand by |

| Standard Name | Translation |
|---|---|
| D_WORD | /Word |

# Commands Dictionary (/site/dict/ccucmds)

The Commands dictionary consists of commands and job list commands and their translations. Your system's version of this file may be slightly different.

Commands may be customized by changing their translation definition. For instance, the default translation of backtime next is `ba n` , as shown in the following commands table. You can change the translation to `back n` or `back next`, or some other translation.

### Commands(/site/dict/ccucmds)

| Standard Name | Translation |
|---|---|
| air | /ai |
| air prompter | /ai p |
| air scrollbox | /ai s |
| backtime | /ba |
| backtime begin | /ba b |
| backtime end | /ba e |
| backtime next | /ba n |
| backtime pause | /ba p |
| blank | /bl |
| bulletin | /bu |
| connect | /co |

| Standard Name | Translation |
|---|---|
| cursor bottom | /cu b |
| cursor down | /cu d |
| cursor left | /cu l |
| cursor right | /cu r |
| cursor top | /cu t |
| cursor up | /cu u |
| define character | /def c |
| define erase | /def e |
| define paragraph | /def p |
| define word | /def w |
| delete character | /del c |
| delete defined | /del d |
| delete recover | /del r |
| destination | /des |
| display | /di |
| display norefresh | /di n |
| display refresh | /di r |
| display story | /di s |
| duplicate | /du |
| enter directory | /e d |
| enter queue | /e q |
| find | /fi |

| Standard Name | Translation |
|---|---|
| flash | /fl |
| get | /ge |
| get all | /ge a |
| get old | /ge o |
| go | /go |
| help | /he |
| hold | /ho |
| insert block | /i b |
| insert character | /i c |
| insert mode | /i m |
| key | /ke |
| kill | /kill |
| load keyboard | /loa k |
| lock | /loc |
| login | /logi |
| login exit | /logi e |
| logout | /logo |
| logout exit | /logo e |
| mail | /ma |
| mail copy | /ma c |
| mail reply | /ma r |
| message | /me |

| Standard Name | Translation |
|---|---|
| message clear | /me c |
| message recall | /me r |
| monitor | /mon |
| move | /mov |
| new | /ne |
| new bottom | /ne b |
| new insert | /ne i |
| new top | /ne t |
| note clear | /no c |
| note recall | /no r |
| note save | /no s |
| order | /or |
| pack | /pac |
| pack all | /pac a |
| page down | /pag d |
| page up | /pag u |
| password | /pas |
| pause | /pau |
| print | /pr |
| publish | /publish |
| quit | /q |
| read | /rea |

| Standard Name | Translation |
|---|---|
| read all | /rea a |
| read old | /rea o |
| receive | /receive |
| remove | /rem |
| replace | /rep |
| save | /sa |
| script | /scri |
| script undo | /scri u |
| scroll bottom | /scro b |
| scroll down | /scro d |
| scroll top | /scro t |
| scroll up | /scro u |
| seek | /see |
| send | /sen |
| split | /sp |
| status clear | /st c |
| tab right | /ta r |
| template recall | /te r |
| template save | /te s |
| time stamp | /ti s |
| transmit | /tr |
| unlock | /unl |

| Standard Name | Translation |
|---|---|
| unsplit | /uns |
| user | /us |
| video define | /vid d |
| video step | /vid s |
| view | /vie |

### Job List Commands

| Standard name | Translation |
|---|---|
| bscan | /bscan |
| bpoll | /bpoll |
| ignore-del | /ignore-del |
| eof | /eof |
| all | /all |
| at | /at |
| every | /every |
| everyentry | /everyentry |
| extension | /extension |
| ignore | /ignore |
| ignore-del | /ignore-del |
| mailto | /mailto |
| number | /number |
| on | /on |

| Standard name | Translation |
|---|---|
| open | /open |
| order | /order |
| poll | /poll |
| priority | /priority |
| put | /put |
| quiet | /quiet |
| scan | /scan |
| sendform | /sendform |
| send-del | /send-del |
| validate | /validate |
| verify | /verify |

# Queues Dictionary (/site/dict/queues)

The Queues dictionary contains names for system queues such as `SYSTEM.KEYBOARDS` and the Dead queue.

Queues in this dictionary are used by functions within iNEWS. For instance, the `seek` command uses whatever queue translation is given to `Q_SEEK`, which is `SYSTEM.SEEK` by default. Like other dictionaries, the standard name is in uppercase and must not be changed. The translation can be in lowercase, but appears in uppercase on the screen. Queue names and their standard translations are shown in the following Queues Dictionary table. The 8-bit codes can be defined using 7-bit sequences.

### Queues Dictionary

| Standard Name | Translation |
|---|---|
| Q_ACCT | /system.account |
| Q_ADDRLIST | /system.teleprinter.addrlist |
| Q_ADDRDIST | /system.teleprinter.distribution |
| Q_CATWORDS | /system.wires.distribution |
| Q_CLIENT_DOS | /system.client.dos |
| Q_CLIENT_COM | /system.client.com |
| Q_CLIENT _REMOTE_SEARCH | /system.client.remote-search |
| Q_CLIENT_COM_VERSIONS | /system.client.com-versions |
| Q_CLIENT_RS_VERSIONS | /system.client.rs-versions |
| Q_CLIENT_WIRE_SERVER | /system.client.wire-server |
| Q_CLIENT_WINDOWS | /system.client.windows |
| Q_CLIENT_VERSIONS | /system.client.versions |
| Q_CONFIGURE | /system.configure |
| Q_DEAD | /dead |
| Q_DESTINATION | /notes |
| Q_FLASH | /wires.advisory.priority |
| Q_FORMS | /system.forms |
| Q_FTS | /system.fts |
| Q_GROUPS | /system.groups |
| Q_HELP | /system.help.terminal |
| Q_HOME | /notes |

| Standard Name | Translation |
|---|---|
| Q_INDEX | /system.index |
| Q_KEYBOARDS | /system.keyboards |
| Q_KEYWORDS | /system.wires.keywords |
| Q_MAIL | /mail |
| Q_MAILOUT | /system.mail.out |
| Q_MAILERROR | /system.mail.error |
| Q_MESSAGE | /system.message |
| Q_MOS_MAP | /system.mos-map |
| Q_MMAP | /system.map |
| Q_MRESOURCE | /system.resource |
| Q_NODEST | /system.unknown |
| Q_PRINTERS | /system.printers |
| Q_PRINTER2 | /system.printers2 |
| Q_REMOTE_SEARCH | /system.remote-search |
| Q_SEARCHTAPE | /system.searchtape |
| Q_SEEK | /system.seek |
| Q_STYLES | /system.styles |
| Q_SERVICE | /system.service |
| Q_SCRIPT | /system.dialogs |
| Q_SPELL | /system.spell |
| Q_TEMPLATES | /system.templates |
| Q_TITLE _ENTRY | /system.title-entry |

| Standard Name | Translation |
|---|---|
| Q_UNKNOWN | /wires.unknown |
| Q_USERROOT | /people |
| Q_WEBACC_FORMS | /system.webforms |
| Q_WEBPUB_FORMS | /system.webforms |

# Words Dictionary (/site/dict/words)

The Words dictionary contains translations for a variety of miscellaneous words used by the system. For instance, words regarding priority (such as flash and silent) or print options (such as story and script) are included.

Standard names are in uppercase and must not be changed. Because many messages in this dictionary are displayed in the upper right corner of active stories and rundowns, keep them short to avoid overwriting portions of the story or rundown. Translations can be uppercase, lowercase, or mixed case.

### Words Dictionary

| Standard Name | Translation |
|---|---|
| **Wire Priorities and Options** | |
| W_FLASH | /FLASH |
| W_BULLETIN | /BULLETIN |
| W_WIRE_FORM | /wires |
| W_URGENT | /URGENT |
| W_SILENT | /SILENT |
| W_ALWAYS | /A |
| W_TRANSMIT | /TRANSMIT |

| Standard Name | Translation |
|---|---|
| **Status Types** | |
| W_HOLD | /HOLD |
| W_LOCKED | /LOCKED |
| W_MAIL | /mail |
| W_READY | /READY |
| W_NEW | /NEW |
| W_WIRE | /WIRE |
| **Special Words for Find** | |
| W_AND | /and |
| W_NOT | /not |
| W_ANDNOT | /andnot |
| W_OR | /or |
| W_NOTREADY | / |
| W_SLOW | /slow |
| W_ALL | /all |
| **Print Command Options** | |
| W_STORY | /story |
| W_SCRIPT | /script |
| W_RUNDOWN | /rundown |
| W_DIRECTORY | /directory |
| W_ON | /on |

| Standard Name | Translation |
|---|---|
| W_DAYS | /SunMonTueWedThuFriSat<br>You can change these translations only once. If you make a mistake, or want to change them again, you must extract the news program from the release CD first. Call iNews Customer Support for assistance in extracting the program. |
| W_MONTHS | /JanFebMarAprMayJunJulAugSepOct NovDec<br>You can change these translations only once. If you make a mistake, or want to change them again, you must extract the news program from the release CD first. Call iNews Customer Support for assistance in extracting the program. |
| W_PAGE | /page |
| **Words Relating to the Seek Server** | |
| W_FAST | /fast |
| W_ACTIVE | /ACTIVE |
| W_DONE | /DONE |
| W_ERROR | |
| W_ABORT | /ABORTED |
| W_MBSPACE | /¡@<br>This token applies to multibyte iNEWS systems only. It must be set to the multibyte space character of the code set being used. |
| W_DELIMITERS | / <sp> “()-.’/ |
| W_PENDING | /PENDING |
| W_RESTRICTED | /restricted |

| Standard Name | Translation |
|---|---|
| W_SLOW | /slow |
| W_BREAK | /= = = = |
| **Words relating to FTS Server** | |
| W_INDEXBASE | /C:/FTS/FTSINDEX |
| W_BINDFTSI | /ftsserver:6100 |
| W_BINDFTSS | ftsserver:6101 |
| W_AGEWEIGHT | /100 |
| W_HITSWEIGHT | /0 |
| W_LOCWEIGHT | /0 |
| **Miscellaneous Options** | |
| W_DEFAULT_FORM | /default_form |
| W_WEBACC_FORM | /access_form |
| W_WEBPUB_FORM | /publish_form |
| W_LOGTYPES | /D |
| W_ERROR | /ERROR (only seek) |
| W_INDXD | /archvd |
| W_DEST | /destination |
| W_QUEUE | /queue |
| W_RESULTS_FORM | /search-results |
| W_START | /ON |
| W_OFF | /OFF |
| W_LOAD | /LOAD |
| W_UNLOAD | /UNLOAD |
| W_YES | /yes |

| Standard Name | Translation |
|---|---|
| W_NO | /no |
| W_GROUP | /group |
| W_ALIAS | /alias |
| W_ANYSTR | /- |
| W_BLANKSTR | /+ |
| W_PAGEBREAK | /pagebreak<br>(used by various programs and identifies the word used to replace the NSML <pb> (pagebreak) tag when converting to SEP. |
| W_RESTRICTED | /restricted<br>(only used by gtraits program and identifies the name used for the restricted user group) |
| W_NEAR | /near |
| W_RUNS | /runs |
| W_STARTMEDIA | /Primary |
| W_CUE_REF | /164 |

*Both W_PENDING and W_BREAK may appear in the Words Dictionary however they are obsolete.*

# Connect Dictionary (/site/dict/doac)

The Connect dictionary contains messages and commands used by the Connect feature.

The standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

## Connect Dictionary

| Standard Name | Translation |
|---|---|
| S_SNPROMPT | /cmd> |
| S_SNCAPON | /Capturing session to |
| S_SNCAPOFF | /Session saved to |
| S_SNPAUSE | /Pause capture |
| S_SNESCAPE | /New escape character |
| S_SNQUIT | /Quitting |
| S_SNCAPERR | /Capture error! |
| S_SNNOQUEUE | /Could not append to queue |
| S_SNCRERR | /Error creating capture story |
| S_SNCLOSED | /Connection closed |
| S_NOSVC | /Unknown service |
| S_NOCAPTURE | /Session not saved to |
| S_SNNOTCAP | /Not capturing |
| S_SNEXPECT | /Failed to get expected string |
| S_NSCRIPT | /Could not open script story |
| S_CONNECT | /CONNECT |
| S_ACCEPT | /ACCEPT |
| S_REJECT | /REJECT |
| S_FINISH | /FINISH |
| S_TERMINAL | /TERMINAL |
| S_RXNET | /RXNET |

| Standard Name | Translation |
|---|---|
| S_RXDNET | /RXDNET |
| S_CCAPTURE | /capture |
| S_CQUIT | /quit |
| S_CPAUSE | /pause |
| S_CSTOP | /stop |
| S_CHELP | /help |
| S_CESCAPE | /escape |
| S_CECHO | /echo |
| S_CHEOL | /heol |
| S_CTYPE | /type |
| S_CWAIT | /wait |
| S_CMESS | /message |
| S_CEXPECT | /expect |
| S_CDELAY | /delay |
| S_CTIMER | /timer |
| S_CPASS | /pass |
| S_CDIAG | /diag |
| S_CMAP | /map |
| S_CMAPIN | /mapin |
| S_CMAPOUT | /mapout |
| S_CAPHLP | /<queue><br>Capture to queue, or continue after a pause |

| Standard Name | Translation |
|---|---|
| S_QUITHLP | /End this connect session |
| S_PAUSEHLP | /Pause capture, but do not close capture story |
| S_STOPHLP | /Stop capturing and close capture story |
| S_HELPHLP | /Show this list of commands |
| S_ESCHLP | /<c> Change escape character to specified character |
| S_HEOLHLP | /Toggle Hard-End-Of-Line on captured data |
| S_ECHOHELP | /Toggle local character echo |
| S_MAPHLP | /<fromchar> <tochar> map input & output |
| S_MAPINHLP | /<fromchar> <tochar> map input |
| S_MAPOUTHLP | /<fromchar> <tochar> map output |
| S_NXNONAME | /Computer not named |
| S_NXNOCONF | /System not configured |
| S_NXNOPTY | /No ptys available |
| S_NXNODEV | /No device available |
| S_KTRANSFER | /transfer |
| S_KCANCELLED | /cancelled |
| S_KDISCARDED | /discarded |
| S_KLOCAL | /L: |
| S_KREMOTE | /R: |
| S_KEOF | /[EOF] |

| Standard Name | Translation |
|---|---|
| S_KEOT | /[EOT] |
| S_KTIMEOUT | /Timed out |
| S_KUNKNOWN | /Unknown packet type |
| S_KIOPENERR | /Input file open error |
| S_KINOTFOUND | /File not found |
| S_KOOPENERR | /Output file open error |
| S_KOWRITERR | /Output file write error |
| S_KOCLOSERR | /Output file close error |

# Telex Dictionary (/site/dict/telex)

The Telex dictionary contains commands and messages used with the telex feature.

### Telex Dictionary

| Standard Name | Translation |
|---|---|
| X_0 | /LINE FREE |
| X_1 | /LINE UNAVAILABLE |
| X_10 | /RING INDICATE |
| X_20 | /CONNECTION MADE |
| X_36 | /INCORRECT AB |
| X_43 | /NO CONNECTION |
| X_50 | /CALL CLEARED |
| X_61 | /INVALID COMMAND |

| Standard Name | Translation |
|---|---|
| X_70 | /SS ABS |
| X_71 | /SS DER |
| X_72 | /SS MON |
| X_73 | /SS NCH |
| X_74 | /SS NP |
| X_75 | /SS NA |
| X_76 | /SS OCC |
| X_77 | /SS NC |
| X_81 | /SS INF |
| X_82 | /SS CIC |
| X_83 | /SS RDI |
| X_DISTRIBUTE | /SYSTEM.TELEX.DISTRIBUTION |
| X_UNKNOWN | /dead |
| X_CMD | /CMD |
| X_SENT | /SENT |
| X_ABORT | /ABORT |
| X_BUSY | /BUSY |
| X_NODIAL | /NODIAL |
| X_NOCNX | /NOCNX |
| X_MORE | /MORE |
| X_RETRY | /RETRY |
| X_FAIL | /FAIL |

| Standard Name | Translation |
|---|---|
| X_MLKUP | /NO LOOKUP ENTRY FOUND |
| X_MMORE | /SENT OK REMAINING SENDS: |
| X_MSENT | /SENT OK |
| X_MRETRY | /NO CONNECTION–RETRIES LEFT: |
| X_MFAIL | /FAILED–NO MORE RETRIES |

# Dial Dictionary (/site/dict/dial)

The Dial dictionary contains messages used with modems.

Like other dictionaries, the standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

### Dial Dictionary

| Standard Name | Translation |
|---|---|
| C_USAGE | /Dial: bad usage |
| C_MDMTO | /Modem timed out |
| C_NOCONS | /Could not open `/dev/console` |
| C_MDMRDY | /Line ready... |
| C_DSYSERR | /Dial: system error |
| C_MDMNOT | /Modem not responding |
| C_MDMNODIAL | /Modem not responding to dial command |
| C_MDMNOHUP | /Modem did not hang up–Check modem |
| C_MDMHUP | /Modem is hung up |

| Standard Name | Translation |
|---|---|
| C_QUITQ | /Do you really want to quit (y/n)? |

# Keyboard Macros Dictionary (/site/dict/keymacros)

The keyboard macros dictionary contains names of keyboard keys for use in keyboard macro definitions for iNEWS NRCS.

Like other dictionaries, the standard name is in uppercase and must not be changed. Translations can be in lowercase, uppercase, or mixed case.

### Keyboard Macros Dictionary

| Standard Name | Translation |
|---|---|
| K_NULL | /null |
| K_F1 | /f1 |
| K_F2 | /f2 |
| K_F3 | /f3 |
| K_F4 | /f4 |
| K_F5 | /f5 |
| K_F6 | /f6 |
| K_F7 | /f7 |
| K_F8 | /f8 |
| K_F9 | /f9 |
| K_F10 | /f10 |
| K_F11 | /f11 |
| K_F12 | /f12 |

| Standard Name | Translation |
|---|---|
| K_KP0 | /kp0 |
| K_KP1 | /kp1 |
| K_KP2 | /kp2 |
| K_KP3 | /kp3 |
| K_KP4 | /kp4 |
| K_KP5 | /kp5 |
| K_KP6 | /kp6 |
| K_KP7 | /kp7 |
| K_KP8 | /kp8 |
| K_KP9 | /kp9 |
| K_INSERT | /insert |
| K_HOME | /home |
| K_PAGEUP | /pageup |
| K_PAGEDOWN | /pagedown |
| K_DELETE | /delete |
| K_END | /end |
| K_UP | /up |
| K_DOWN | /down |
| K_LEFT | /left |
| K_RIGHT | /right |
| K_SHIFT | /shift |
| K_CTRL | /ctrl |

| Standard Name | Translation |
|---|---|
| K_ALT | /alt |
| K_TAB | /tab |
| K_ESC | /esc |
| K_BACKSPACE | /backspace |
| K_ENTER | /enter |
| K_PAUSE | /pause |
| K_REPEAT | /repeat |
| K_SPACE | /space |

# Printer Messages Dictionary (/site/dict/printmsgs)

The Printer Messages dictionary contains error messages generated by the Sony barcode printer.

The standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

### Printer Messages Dictionary

| Standard Name | Translation |
|---|---|
| M_TAPEIDREQ | /Tape ID required |
| M_INVTAPEID | /Invalid tape ID |
| M_DUREXCEED | /Duration time exceeded |
| M_DURTOOSHORT | /Duration time too short |
| M_INVTIMECODE | /Invalid timecode |
| M_EOMREQ | /EOM required |

| Standard Name | Translation |
|---|---|
| M_SOMREQ | /SOM required |
| M_INVCHARS | /Tape ID or title has invalid characters |
| M_EOMSTMISSING | /EOM Style missing from profile |
| M_TAPEIDLEN | /Tape ID must be between 3-8 characters long |
| M_PAGESTRLEN | /Page string too long |
| M_SOMEXCEEDS | /SOM timecode exceeds EOM timecode |
| M_UNKERR | /Unknown print error |
| A_CMDDEL | /* |

# Case-shifting Dictionary (/site/dict/shift)

The Case-shifting dictionary maps lowercase characters to their uppercase counterparts and vice versa. iNEWS shifts the case of a character according to its decimal value in a standard character conversion table.

The dictionary has two parts:

- The first part, labeled with the keyword tolower, maps decimal values of uppercase characters to the decimal values of their lowercase counterparts.
- The second part, labeled with the keyword toupper, maps decimal values of lowercase characters to decimal values of their uppercase counterparts.

In the default dictionary shipped with iNEWS newsroom computer system, a character at a decimal position in the range on the left of the arrow (->) shifts to the character at the corresponding decimal position in the range on

the right. For instance, the character at decimal position 65 (A) is mapped to the character at decimal position 97 (a); the character at decimal position 66 (B) shifts to the character at decimal position 98 (b); and so on:

```
;
tolower
 65 -  90 ->  97 - 122      ; A - Z -> a - z
192 - 207 -> 224 - 239
209 - 221 -> 241 - 253
end

toupper
 97 - 122 ->  65 -  90      ; a - z -> A - Z
224 - 239 -> 192 - 207
241 - 253 -> 209 - 221
end
```

The character-conversion table the system uses depends on the interface you are using.

- If you are using the DOS, the character-conversion table is based on the DEC Multinational Character Set (MCS).
- If you are using the iNEWS Graphic User Interface (GUI), the conversion table is based on the ISO standard for multinational characters.

If character mappings specified in these standard character-conversion tables are not appropriate for the language you are using, edit the `/site/dict/shift` file to remap character conversions. You can map ranges of values (as shown in the default dictionary file) or you can map values one by one, if necessary.

When editing the dictionary file, follow these guidelines:

- Ensure all keywords (tolower, toupper, end) in the dictionary file remain in lowercase.
- Specify all character-conversions in terms of the characters' decimal values in the conversion table.
- Do not specify a value higher than 255.

- Any characters not mapped in the dictionary file remain the same when shifted.
- The system ignores blank lines in the dictionary file and any characters following a semicolon (;).

After you edit the dictionary file, run the `makeshift` console command in maintenance mode during installation to prepare the Case-shifting dictionary for use by the iNEWS newsroom computer system. For more information, call Avid Customer Support.

If you map a character to more than one value, the system displays a warning when you type the `makeshift` command, but uses the last character mapping in the file.

# MCS Dictionary (/site/dict/mcs)

The MCS dictionarycontains the following:

- Device Types Used by Monitor Servers and Drivers
- Special Strings Recognized by the Monitor Server
- Error Messages for the Monitor Server
- Error Messages Shared by Monitor Servers and Drivers
- Driver Errors Sent to MCT Only
- Driver Errors Sent to MCT and Possibly to Console & Mon User
- Driver Errors Sent to User Only
- Status Reported in Device Status Field

The standard name is in uppercase and must not be changed. The translation can be in lowercase, uppercase, or mixed case.

### Device Types Used by Monitor Servers and Drivers

| Standard Name | Translation |
|---|---|
| A_CART | /CART |

| Standard Name | Translation |
|---|---|
| A_CG | /CG |
| A_MOS | /MOS |
| A_SS | /SS |
| A_USRDEF | /USRDEF |
| A_CART | /CART |

### Special Strings Recognized by the Monitor Server

| Standard Name | Translation |
|---|---|
| A_BREAK | /==== |
| A_CHANNELOPEN | /[ |
| A_CHANNELCLOSE | /] |
| A_CMDDEL | /* |
| A_COMDEL | /; |
| A_CURRENT | /<<<< |
| A_FLDDEL | / |
| A_LOCALHYPHEN | /- |
| A_NOUPDT | /noupdate |
| A_UPDATE | /update |

### Error Messages for the Monitor Server

| Standard Name | Translation |
|---|---|
| A_MBINVL | /Invalid mailbox |

| Standard Name | Translation |
|---|---|
| A_NOTINMAP | /Queue not in map |
| A_XMAXSTORY | /Exceeded max # stories, monitor exiting |
| A_MAPOPENERR | /Map story open error |
| A_RESOPENERR | /Resource story open error |
| A_NOSERVER | /Server cannot execute |
| A_BADCG | /Bad Map line: CG device requires range |
| A_BADMCT | /Bad Map line: Invalid MCT template |
| A_BADMAPARG | /Bad Map line: Invalid arg count |
| A_BADMAPUPDT | /Bad Map line: Expected update arg |
| A_BADMAPQTME | /Bad Map line: Invalid time value |
| A_BADMAPDUPE | /Bad Map line: Duplicate device |
| A_BADMAPDRV | /Bad Map line: Invalid drive specification |
| A_BADMAPINVCHAN | /Bad Map line: Invalid video channel assignment policy |
| A_BADMAPBADMOS | /Bad Map line: Unrecognized MOS parameter |
| A_BADMAPNOMOS | /Bad Map line: MOS device not assigned to MOS Server |
| A_BADMAPBADSVR | /Bad Map line: device-server type mismatch |
| A_BADMOSITEM | /Invalid MOS item |
| A_BADDUPENTRY | /Bad Resource line: Duplicate style entry |
| A_BADCGLINES | /Bad Resource line: Invalid # CG fields |
| A_BADRESARG | /Bad Resource line: Invalid arg count |
| A_BADCGTMPL | /Bad Resource line: Invalid CG template |

| Standard Name | Translation |
|---|---|
| A_RESERRORS | /Following errors found in resource story |
| A_INVLDEV | /Invalid device specification |
| A_INVLSTYLE | /Invalid style |
| A_INVLADD | /Invalid address |
| A_INVLARG | /Missing argument |
| A_XMAXADD | /Exceeded address range |
| A_NOFORM | /Cannot access form |
| A_SRTERR | /Error sorting queue |
| A_DELERR | /Cannot delete from |
| A_DIRERR | /Directory open error |
| A_QUEERR | /Queue open error |
| A_QUEAPPERR | /Queue append error |
| A_STRYERR | /Story open error |
| A_STRYCRT | /Story create error |
| A_STRYWRT | /Story write error |
| A_DRVNOTINMAP | /Driver not in Map |
| A_DRVNCONFIG | /Download failed: driver not configured |
| A_DRVNOCOMM | /Network connect failed |
| A_DEVBUSY | /Refused connection request |
| A_NOSSADDR | /No SS address specified |

| Standard Name | Translation |
|---|---|
| A_MULSSADDR | /Multiple SS addresses specified |
| A_MONEXIT | /Monitor exiting |
| A_QUITTIME | /Quite Time reached |
| A_MUSTUNLOAD | /Must first unload |
| A_READY | /OK |

### Error Messages Shared by Monitor Servers and Drivers

| Standard Name | Translation |
|---|---|
| A_XMAX | /Exceeded max # events |
| A_NOMEM | /Out of memory |

### Driver Errors Sent to MCT Only

| Standard Name | Translation |
|---|---|
| A_NOHOST | /Disconnected from host |

### Driver Errors Sent to MCT and Possibly to Console & Mon User

| Standard Name | Translation |
|---|---|
| A_NODEV | /Device error |
| A_NODEVCOMM | /Device communication failed |

### Driver Errors Sent to User Only

| Standard Name | Translation |
|---|---|
| A_LOADDONE | /Download complete |
| A_LOADERR | /Errors in download |
| A_DRIVEREXIT | /Driver exiting |
| A_CANOTCUED | /Tape not cued |

### Status Reported in Device Status Field

| Standard Name | Translation |
|---|---|
| A_EVERR | /ERROR |
| A_CAFRZ | /FREEZE |
| A_CATREL | /TREL |
| A_CATHRD | /THREAD |
| A_CACUING | /CUEING |
| A_CACUED | /CUED |
| A_CANOTAPE | /N/L |
| A_CABIN | /BIN |
| A_CAPLAY | /PLAY |
| A_CAPAUSE | /PAUSED |
| A_CAREW | /REWIND |
| A_CAEJECT | /EJECT |
| A_CAINCMPLT | /TRANSFER |

# Appendix D

# PCU Reference

This appendix contains the following main sections:

- Overview
- PCUs

# Overview

A Peripheral Controller Unit (PCU) is a Personal Computer (PC) connected between an iNEWS Server and one or more serial devices, such as printers and wire services. The PCU expands the available serial ports and relieves the server of routine communication with the devices. The PCU is an open system. The operating system is Microsoft DOS and a standard public domain TCP/IP network stack is used.

No server should be connected to more than half the number of PCUs it is capable of supporting. This ensures that if one server fails, the surviving server(s) have enough reserve capacity to run the failed server's PCUs.

Divide your system's PCUs evenly among your system's servers. This yields maximum performance possible from each server. If a server fails, reconfigure the failed server's PCUs to the surviving servers until the failed server is repaired.

# PCUs

There are two kinds of PCUs: older models in rack-mount cases and newer models in standard PC cases. The latter is referred to as a PC-PCU. A PC-PCU can be an ordinary PC with special additional hardware to give it more serial ports. This section describes the rack-mount models.

*For more information about PC-PCUs, see iNEWS Newsroom Computer System Setup and Configuration Manual.*

A PCU is essentially a DOS PC. A PCU's front panel contains one or two LED displays and reset switches. The reset switches allow you to clear the PCU's memory prior to restarting it. LEDs display the PCU's device number or current status. This section describes parts of a PCU with which you need to be familiar to do normal system maintenance work. It also describes how to reset the PCU, how to connect the PCU to a server, and how to connect devices to the PCU.

*Avid no longer supports CCUs. If your system has existing CCUs that have been upgraded to PCUs, such devices must be configured as PCUs.*

## Resetting a PCU

Occasionally, there may be a need to reset a PCU to clear its memory so that you can load a fresh set of programs. To do this, press the Reset button and let go. The PCU tests its hardware and boots up to a point where you can restart it from the console.

A PCU built to resemble a CCU has one or two LED displays with four digits each. It uses the LED display to indicate whether it is running normally, has just been reset, or has detected an error. As long as the PCU is running normally, the LED displays the PCU's device number.

If the LED goes out, the PCU may have lost power, so make certain the power switch is in the "on" position and the PCU's power cord is firmly attached to the PCU and power supply. PCUs produced in regular PC housings do not have LED displays.

As the PCU tests itself, it shows a number of status codes on the LED display. When DOS boots on the PCU, it displays "8000." If the PCU passes all its diagnostic tests and receives an IP address from a bootp server, it displays "0000," indicating that it has successfully completed the self test and is ready to be restarted. If the PCU continues to display "8000," there is a network or bootp problem.

A PCU displays "8888" after receiving a reset request from the host server. After it has been restarted and is operational, it displays its device number (the device number is not divided by 10).

## PCU LED Displays

There are standard codes that can appear on the PCU's LED display. The 8000 codes are typically shown during a normal boot of the PCU, while the 9000 codes represent errors. Contact Avid Customer Support for assistance if one of your PCUs displays an error code after being restarted. The following PCU LED Code table lists the codes and their meanings.

### PCU LED Codes

| Code | Meaning |
|---|---|
| 0000 | Network initialization complete |
| 8000 | Boot started |
| 8002 | Connected to host server for download of PCU OS |
| 8003 | PCU OS download in progress |
| 8004 | PCU OS downloaded |
| 8005 | Start received |
| 8006 | Invoking PCU OS |
| 8100 | PCU OS started |
| 8101 | PCU OS network initialized |
| 8102 | PCU OS fully operational |
| 8888 | Reset received |
| 9001 | Error, execv failed |
| 9002 | Error, file could not be opened |
| 9003 | Error, no data downloaded |
| 9004 | Error, cannot establish socket connection |
| 9005 | Error, read on socket failed |
| 9006 | Error, receive on socket failed |
| 9007 | Error, send on socket failed |
| 9010 | Memory management—bad free |
| 9011 | Memory management—unknown free |
| 9012 | Memory management—no space |

## Connecting Devices to a PCU

You use the I/O ports to connect serial devices, such as printers and wires, to the PCU. These ports are numbered 1 through 8. Use these port numbers in combination with the PCU's device number when assigning device numbers to devices you connect to the PCU. These ports are Data Terminal Equipment (DTE) RS-232 serial ports. To connect a device to one of these ports, use a cable built to carry signals the device and PCU need, in a way they expect.

### Locating Ports

The back panel of a PCU built in a CCU III housing contains eight I/O ports, a host port, a power receptacle, an Ethernet connector, and a power switch.

The power switch controls power to the PCU and should always be in the on position. The power receptacle is where the power cord attaches to the PCU. The host port is used to attach a diagnostic workstation to the PCU, if necessary.

A PCU in a standard PC case looks similar to this, except that eight serial I/O ports are on a serial card in one of the PC's expansion slots.

## Connecting a PCU to the System

A PCU has an Ethernet connector. Use this connector to attach the PCU to the Ethernet network so it can communicate with its server.

# Appendix E

# Character Maps

This appendix contains character mapping code tables used by wire programs. The value set in the “bits” wire profile option determines which of these tables the wire program uses to “translate” stories it receives from a wire service. Unless otherwise indicated, the tables contain, in the first column, decimal numbers sent by the wire service. The other columns contain characters represented by those numbers, descriptions of the characters, and mapped values.

Also included in this appendix are examples of character mapping for Arabic wires.

This section contains the following:

- ASCII (7-bit) Character Set
- IBM Character Set
- dbrestore Character Map (Arabic)
- Sample Arabic Wire Profile

## ASCII (7-bit) Character Set

The following ASCII (7 bit) table is an identity mapping, where the wire value equals the mapped value, with one exception—the wire value of 124 is mapped to 0. The other 127 values are preserved.

**ASCII (7 bit)**.

| Value | Mapped Alias | Description | Mapped Value |
|---|---|---|---|
| 000 | <nul> | null | 000 |
| 001 | <soh> | start of header | 001 |
| 002 | <stx> | start of text | 002 |
| 003 | <etx> | end of text | 003 |
| 004 | <eot> | end of transmission | 004 |
| 005 | <enq> | enquiry | 005 |
| 006 | <ack> | acknowledgment | 006 |
| 007 | <bel> | bell | 007 |
| 008 | <bs> | back space | 008 |
| 009 | <ht> | horizontal tab | 009 |
| 010 | <lf> | line feed | 010 |
| 011 | <vt> | vertical tab | 011 |
| 012 | <ff> | form feed | 012 |
| 013 | <cr> | carriage return | 013 |
| 014 | <so> | shift out | 014 |
| 015 | <si> | shift in | 015 |
| 016 | <dle> | data link escape | 016 |
| 017 | <dcl> | device control 1 | 017 |
| 018 | <dc2> | device control 2 | 018 |
| 019 | <dc3> | device control 3 | 019 |
| 020 | <dc4> | device control 4 | 020 |
| 021 | <nak> | negative acknowledge | 021 |
| 022 | <syn> | synchronous idle | 022 |

| Value | Mapped Alias | Description | Mapped Value |
|---|---|---|---|
| 023 | <etb> | end of transmission | 023 |
| 024 | <can> | cancel | 024 |
| 025 |  | end of medium | 025 |
| 026 | <sub> | substitute | 026 |
| 027 | <esc> | escape | 027 |
| 028 | <fs> | file separator | 028 |
| 029 | <gs> | group separator | 029 |
| 030 | <rs> | record separator | 030 |
| 031 | <us> | unit separator | 031 |
| 032 | <sp> | space | 032 |
| 033 | ! | | 033 |
| 034 | “ | | 034 |
| 035 | # | | 035 |
| 036 | $ | | 036 |
| 037 | % | | 037 |
| 038 | & | | 038 |
| 039 | ' | quote | 039 |
| 040 | ( | | 040 |
| 041 | ) | | 041 |
| 042 | * | | 042 |
| 043 | + | | 043 |
| 044 | , | comma | 044 |
| 045 | - | | 045 |
| 046 | . | period | 046 |

| Value | Mapped Alias | Description | Mapped Value |
|---|---|---|---|
| 047 | / | | 047 |
| 048 | 0 | | 048 |
| 049 | 1 | | 049 |
| 050 | 2 | | 050 |
| 051 | 3 | | 051 |
| 052 | 4 | | 052 |
| 053 | 5 | | 053 |
| 054 | 6 | | 054 |
| 055 | 7 | | 055 |
| 056 | 8 | | 056 |
| 057 | 9 | | 057 |
| 058 | : | colon | 058 |
| 059 | <sc> | | 059 |
| 060 | < | | 060 |
| 061 | = | | 061 |
| 062 | > | | 062 |
| 063 | ? | | 063 |
| 064 | @ | | 064 |
| 065 | A | | 065 |
| 066 | B | | 066 |
| 067 | C | | 067 |
| 068 | D | | 068 |
| 069 | E | | 069 |
| 070 | F | | 070 |

| Value | Mapped Alias | Description | Mapped Value |
|---|---|---|---|
| 071 | G | | 071 |
| 072 | H | | 072 |
| 073 | I | | 073 |
| 074 | J | | 074 |
| 075 | K | | 075 |
| 076 | L | | 076 |
| 077 | M | | 077 |
| 078 | N | | 078 |
| 079 | O | | 079 |
| 080 | P | | 080 |
| 081 | Q | | 081 |
| 082 | R | | 082 |
| 083 | S | | 083 |
| 084 | T | | 084 |
| 085 | U | | 085 |
| 086 | V | | 086 |
| 087 | W | | 087 |
| 088 | X | | 088 |
| 089 | Y | | 089 |
| 090 | Z | | 090 |
| 091 | [ | | 091 |
| 092 | \ or<bks> | | 092 |
| 093 | ] | | 093 |
| 094 | ^ | | 094 |

| Value | Mapped Alias | Description | Mapped Value |
|---|---|---|---|
| 095 | _ | underscore | 095 |
| 096 | ` | | 096 |
| 097 | a | | 097 |
| 098 | b | | 098 |
| 099 | c | | 099 |
| 100 | d | | 100 |
| 101 | e | | 101 |
| 102 | f | | 102 |
| 103 | g | | 103 |
| 104 | h | | 104 |
| 105 | i | | 105 |
| 106 | j | | 106 |
| 107 | k | | 107 |
| 108 | l | | 108 |
| 109 | m | | 109 |
| 110 | n | | 110 |
| 111 | o | | 111 |
| 112 | p | | 112 |
| 113 | q | | 113 |
| 114 | r | | 114 |
| 115 | s | | 115 |
| 116 | t | | 116 |
| 117 | u | | 117 |
| 118 | v | | 118 |

| Value | Mapped Alias | Description | Mapped Value |
|---|---|---|---|
| 119 | w | | 119 |
| 120 | x | | 120 |
| 121 | y | | 121 |
| 122 | z | | 122 |
| 123 | { | | 123 |
| 124 | <nul> | | 0 |
| 125 | } | | 125 |
| 126 | ~ | | 126 |
| 127 | <del> | Delete | 127 |

# IBM Character Set

The IBM character set is the same as the ASCII 7-bit character set for the first 127 characters. For the remaining characters, see the following IBM Character table.

### IBM Character

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 128 | 80 | Ç | 190 | C7 |
| 129 | 81 | ü | 252 | FC |
| 130 | 82 | é | 233 | E9 |
| 131 | 83 | â | 226 | E2 |
| 132 | 84 | ä | 228 | E4 |
| 133 | 85 | à | 224 | E0 |
| 134 | 86 | å | 229 | E5 |

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 135 | 87 | ç | 231 | E7 |
| 136 | 88 | ê | 234 | EA |
| 137 | 89 | ë | 235 | EB |
| 138 | 8A | è | 232 | E8 |
| 139 | 8B | ï | 239 | EF |
| 140 | 8C | î | 238 | EE |
| 141 | 8D | ì | 236 | EC |
| 142 | 8E | Ä | 196 | C4 |
| 143 | 8F | Å | 197 | C5 |
| 144 | 90 | É | 201 | C9 |
| 145 | 91 | æ | 230 | E6 |
| 146 | 92 | Æ | 198 | C6 |
| 147 | 93 | ô | 244 | F4 |
| 148 | 94 | ö | 246 | F6 |
| 149 | 95 | ò | 242 | F2 |
| 150 | 96 | û | 251 | FB |
| 151 | 97 | ù | 249 | F9 |
| 152 | 98 | ÿ | 253 | FD |
| 153 | 99 | Ö | 214 | D6 |
| 154 | 9A | Ü | 220 | DC |
| 155 | 9B | ¢ | 162 | A2 |
| 156 | 9C | £ | 163 | A3 |
| 157 | 9D | ¥ | 165 | A5 |

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 158 | 9E | <nul> | 0 | 0 |
| 159 | 9F | <nul> | 0 | 0 |
| 160 | A0 | á | 225 | E1 |
| 161 | A1 | í | 237 | ED |
| 162 | A2 | ó | 243 | F3 |
| 163 | A3 | ú | 250 | FA |
| 164 | A4 | ñ | 241 | F1 |
| 165 | A5 | Ñ | 209 | D1 |
| 166 | A6 | ª | 170 | AA |
| 167 | A7 | º | 186 | BA |
| 168 | A8 | <nul> | 0 | 0 |
| 169 | A9 | <nul> | 0 | 0 |
| 170 | AA | <nul> | 0 | 0 |
| 171 | AB | <nul> | 0 | 0 |
| 172 | AC | <nul> | 0 | 0 |
| 173 | AD | <nul> | 0 | 0 |
| 174 | AE | <nul> | 0 | 0 |
| 175 | AF | <nul> | 0 | 0 |
| 176 | B0 | <nul> | 0 | 0 |
| 177 | B1 | <nul> | 0 | 0 |
| 178 | B2 | <nul> | 0 | 0 |
| 179 | B3 | <nul> | 0 | 0 |
| 180 | B4 | <nul> | 0 | 0 |

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 181 | B5 | <nul> | 0 | 0 |
| 182 | B6 | <nul> | 0 | 0 |
| 183 | B7 | <nul> | 0 | 0 |
| 184 | B8 | <nul> | 0 | 0 |
| 185 | B9 | <nul> | 0 | 0 |
| 186 | BA | <nul> | 0 | 0 |
| 187 | BB | <nul> | 0 | 0 |
| 188 | BC | <nul> | 0 | 0 |
| 189 | BD | <nul> | 0 | 0 |
| 190 | BE | <nul> | 0 | 0 |
| 191 | BF | <nul> | 0 | 0 |
| 192 | C0 | <nul> | 0 | 0 |
| 193 | C1 | <nul> | 0 | 0 |
| 194 | C2 | <nul> | 0 | 0 |
| 195 | C3 | <nul> | 0 | 0 |
| 196 | C4 | <nul> | 0 | 0 |
| 197 | C5 | <nul> | 0 | 0 |
| 198 | C6 | <nul> | 0 | 0 |
| 199 | C7 | <nul> | 0 | 0 |
| 200 | C8 | <nul> | 0 | 0 |
| 201 | C9 | <nul> | 0 | 0 |
| 202 | CA | <nul> | 0 | 0 |
| 203 | CB | <nul> | 0 | 0 |

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 204 | CC | <nul> | 0 | 0 |
| 205 | CD | <nul> | 0 | 0 |
| 206 | CE | <nul> | 0 | 0 |
| 207 | CF | <nul> | 0 | 0 |
| 208 | D0 | <nul> | 0 | 0 |
| 209 | D1 | <nul> | 0 | 0 |
| 210 | D2 | <nul> | 0 | 0 |
| 211 | D3 | <nul> | 0 | 0 |
| 212 | D4 | <nul> | 0 | 0 |
| 213 | D5 | <nul> | 0 | 0 |
| 214 | D6 | <nul> | 0 | 0 |
| 215 | D7 | <nul> | 0 | 0 |
| 216 | D8 | <nul> | 0 | 0 |
| 217 | D9 | <nul> | 0 | 0 |
| 218 | DA | <nul> | 0 | 0 |
| 219 | DB | <nul> | 0 | 0 |
| 220 | DC | <nul> | 0 | 0 |
| 221 | DD | <nul> | 0 | 0 |
| 222 | DE | <nul> | 0 | 0 |
| 223 | DF | <nul> | 0 | 0 |
| 224 | E0 | <nul> | 0 | 0 |
| 225 | E1 | ß | 223 | DF |
| 226 | E2 | <nul> | 0 | 0 |

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 227 | E3 | <nul> | 0 | 0 |
| 228 | E4 | <nul> | 0 | 0 |
| 229 | E5 | <nul> | 0 | 0 |
| 230 | E6 | µ | 185 | B5 |
| 231 | E7 | <nul> | 0 | 0 |
| 232 | E8 | <nul> | 0 | 0 |
| 233 | E9 | <nul> | 0 | 0 |
| 234 | EA | <nul> | 0 | 0 |
| 235 | EB | <nul> | 0 | 0 |
| 236 | EC | <nul> | 0 | 0 |
| 237 | ED | Ø | 248 | F8 |
| 238 | EE | <nul> | 0 | 0 |
| 239 | EF | <nul> | 0 | 0 |
| 240 | F0 | <nul> | 0 | 0 |
| 241 | F1 | ± | 177 | B1 |
| 242 | F2 | <nul> | 0 | 0 |
| 243 | F3 | <nul> | 0 | 0 |
| 244 | F4 | <nul> | 0 | 0 |
| 245 | F5 | <nul> | 0 | 0 |
| 246 | F6 | <nul> | 0 | 0 |
| 247 | F7 | <nul> | 0 | 0 |
| 248 | F8 | ° | 176 | B0 |
| 249 | F9 | <nul> | 0 | 0 |

| Value | Wire Value HEX | Mapped Alias | Mapped Value | Mapped Value HEX |
|---|---|---|---|---|
| 250 | FA | <nul> | 0 | 0 |
| 251 | FB | <nul> | 0 | 0 |
| 252 | FC | <nul> | 0 | 0 |
| 253 | FD | [2] | 178 | B2 |
| 254 | FE | <nul> | 0 | 0 |
| 255 | FF | <nul> | 0 | 0 |

# dbrestore Character Map (Arabic)

The following is an example of a `dbrestore` Character map—in this case, converting characters from ASMO 449+ to Arabic Windows 1256.

```
#this file maps from ASMO 449+ to MS 1256 code page
130 -> 233              ;e acute
131 -> 226              ;a circumflex
133 -> 224              ;a grave
135 -> 231              ;c cidilla
136 -> 234              ;e circumflex
137 -> 235              ;e umlaut
138 -> 232              ;e grave
139 -> 239              ;i umlaut
140 -> 238              ;i circumflex
147 -> 244              ;o circumflex
150 -> 251              ;u circumflex
151 -> 249              ;u grave
172 -> 161              ;arabic comma
174 -.>171              ;left double angle quote
175 -> 187              ;right double angle quote
187 -.>186              ;arabic semi colon
215 - 218 -> 216 - 219  ;tah,zah,ain,ghain
224 - 227 -> 220 - 223  ;tatweel,feh,qaf,kaf
228 -> 225              ;lam
229 - 232 -> 227 - 230  ;meem, noon, heh, waw
233 - 234 -> 236 - 237  ;alef maksura, yeh
235 - 238 -> 240 - 243 ;fathatan,dammatan,kasratan,fatha
239 - 240 -> 245 - 246  ;damma, kasra
241 -> 248              ;shadda
242 ->250               ;sukun
246 -> 161              ;arabic comma
247 => 225 194          ;lam alef with madda above
248 => 225 195          ;lam alef with hamza above
249 => 225 197          ;lam alef with hamza below
250 => 225 199          ;lam alef
# that s all
```

# Sample Arabic Wire Profile

The following is an example of a wire profile for an Arabic wire service, using the map and reverse wire profile options.

```
wire    11      300-5   dummy   MIN   -   -   ;MINA service

# cat /site/wires/11
; MENA WIRE
;iNEWS 1.2 MS Codepage 1256

form    wire-arabic
bits            5
flags           pblines
start
<212><237><212><237><128><LF><128><LF><128><LF><128><LF>
<128><LF><128><LF>
reverse 0123456789 /,.
end             <228><228><228>
slug            30
accent          <bs>
form wire
flags   addheol

map                <00>    _<bs>
map             <01>    <216>
map             <02>    <CR><LF>
map             <03>    <213>
; 4 is space
map             <05>    <198>
map             <06>    <200>
map             <07>    <203>
map             <08>    <223>
map             <09>    <222>
map             <10>    <207>
map             <11>    <229>
map             <12>    <228>
map             <13>    <221>
map             <14>    <237>
map             <15>    <205>
map             <16>    <230>
```

```
map             <17>    <212>
map             <18>    <204>
map             <19>    <225><199>
map             <20>    <218>
map             <21>    <209>
map             <22>    <195>
map             <23>    <225>
map             <24>    <202>
map             <25>    <199>
map             <26>    <218>
;27 is figure shift
map             <28>    <227>
map             <29>    <211>
map             <30>    <199><225>
;31 is letter shift
map                <32>    _<bs>
map             <33>    <55>
map             <34>    <CR><LF>
;34 is new line
map             <35>    <213>
;36 is space
map             <37>    <236>
map             <38>    <50>
map             <39>    <51>
map             <40>    <223>
map             <41>    <222>
map             <42>    <54>
map             <43>    <229>
map             <44>    <228>
map             <45>    <201>
map             <46>    <236>
map             <47>    <205>
map             <48>    <53>
map             <49>    <212>
map             <50>    <204>
map             <51>    <56>
map             <52>    <47>
map             <53>    <52>
map             <54>    <48>
map             <55>    <57>
```

```
map             <56>    <49>
map             <57>    <193>
map             <58>    <218>
;59 is figure shift
map             <60>    <227>
map             <61>    <211>
map             <62>    <225>

;63 is letter shift
```

# Appendix F

# Environment Variables

Some features in the Avid iNEWS newsroom computer system require the system administrator to set up environment variables in the Registry of the workstations. The person responsible for setting up these variables should have a good understanding of Windows-based operating systems, and the Registry Editor program. This appendix includes the following sections:

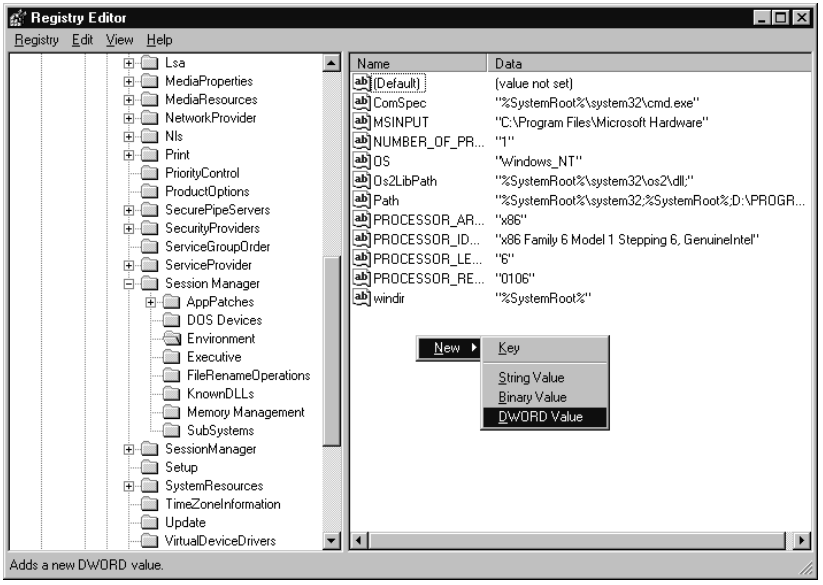
- Registry Editor
- Environment Variables (Registry Values)
- Environmental Variables for Servers

# Registry Editor

The Registry Editor is used to create and define environment variables (Registry values) at each workstation.Environment Variables

**To access the Registry Editor:**

1. Click the Start button on the Windows Taskbar.
2. Select the Run option.
3. Type regedit in the dialog box that appears.
4. The Registry Editor window appears.



All iNEWS newsroom computer system environment variables are set up and stored in the same location on each workstation. After opening the Registry Editor window, navigate to the following folder (also called a key).

```
HKEY_LOCAL_MACHINE\
     SYSTEM\
     CurrentControlSet\
     Control\
     Session Manager\
     Environment
```

*On workstations running the Windows NT-based operating system, there are two keys with similar names:SessionManager and Session Manager. The one called Session Manager (with a space between the two words) must be used*

# Environment Variables (Registry Values)

Environment variables or registry values are sometimes required to set up certain iNEWS features at various iNEWS Workstations. Environment variables are located and defined in the Registry of iNEWS Workstations—that is, Windows-based PCs running the Client software. The following variables are covered in this section:

- CCColor
- DestinationOrder
- MailLookup
- MsgMailAlert
- PIColor
- ShowTimingBar
- SyncToServer
- Delete_Notify

The following sections explain how to can set up environment variables by editing the Registry using the Registry Editor.

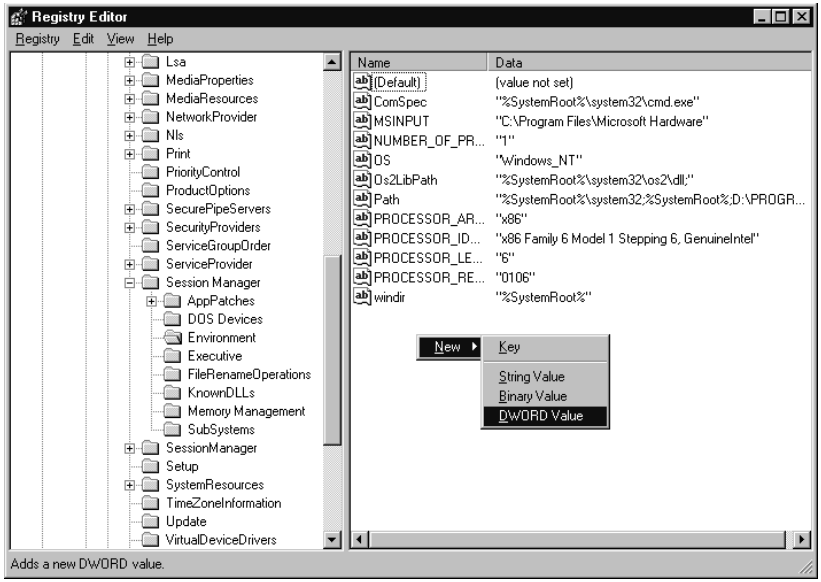
*Self-importing files, called reg files, can be executed to automatically import envionment variable information into the Registry. These files with their exported registry keys, can be used on PCs running Windows NT - based operating systems. For more information on how to obtain and use these reg files, contact Avid Customer Support.*

## CCColor

An individual workstation can have its closed captioning text color changed via an environment variable called CCColor. If no environment variable exists, then the default color of green is used.

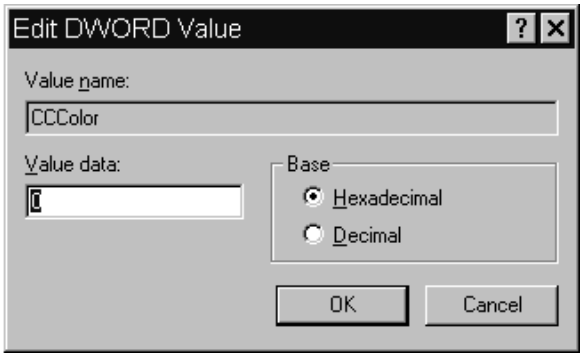
**To change the closed captioning text color:**

1. Open Registry Editor. See “To access the Registry Editor:” on page 274 for more information.
2. Navigate to the Environment key and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.



4. Select the DWORD Value option to create and define a new Registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: CCColor.
6. Press Enter.
7. To set the Value data option:
   - Right-click on the CCColor value.
   - Choose Modify from the pop-up menu.

   The Edit DWORD Value dialog box appears




   - Set the Value data using the following hexadecimal format:

     `0x00RRGGBB`,

     where RR, GG, BB are two bytes for each color.

*The leftmost two bytes (00) are not used. Also, If the CCColor has its value set to zero (0), the closed captioning text will be black because zero corresponds to the color Black.*

   - Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window.

See “RGB Hexadecimal Color Chart” on page 283 for more information on possible colors used in this environment variable

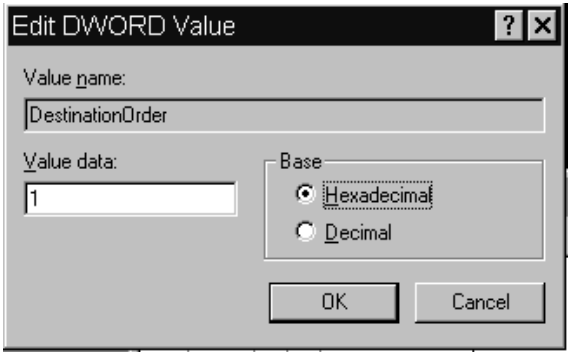
## DestinationOrder

Enabling the destination order feature ensures the user's Home location is always the top item in the Destination queue list. For instance, when you duplicate a story to another queue, the user's Home location will always be the top item in the list. It also ensures the user's Destination location is the second item in the list.

**To enable the destination order feature, do the following:**

1. Open Registry Editor. See "To access the Registry Editor:" on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **DestinationOrder**.
6. Press Enter.
7. To set the Value data option:
   - Right-click on the `DestinationOrder` value.
   - Choose Modify from the pop-up menu.

     The Edit DWORD Value dialog appears.



- Set the Value data. Type `0` (zero) to disable the destination order feature, or `1` to enable it.

*Any number other than `1` turns DestinationOrder off and back to its default behavior, which is to always display the last visited queue/folder as the top item in the destination list.*

- Click OK to save the setting and close the dialog box.

8. Close the Registry Editor window.

## MailLookup

The iNEWS newsroom computer system provides users with an e-mail addressee name lookup feature. When used, all groups, aliases, and users that partially match characters in the To: or CC: fields are displayed in a Check Name dialog box for user selection. This is the default behavior. However, system administrators can set an environment variable that defines which matches are displayed for selection in the Check Name dialog box. Consequently, this allows system administrators to hide any groups that exist in the system for reasons other than e-mail purposes.

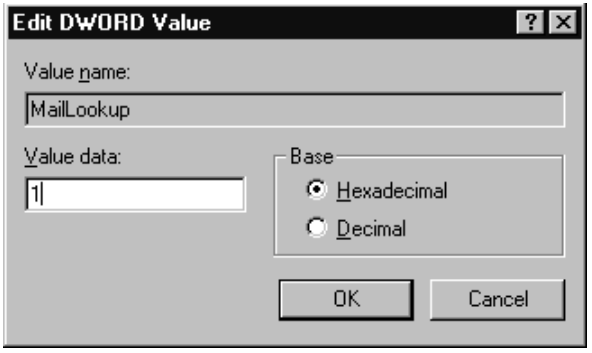
*This environment variable must be created and defined in the Registry Editor at each workstation. Default behavior is used at workstations where the environment variable is not defined.*

**To set the environment variable:**

1. Open Registry Editor. See “To access the Registry Editor:” on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **MailLookup.**
6. Press Enter.
7. To set the Value data option:

- Right-click on the `MailLookup` value.
- Choose Modify from the pop-up menu.

  The Edit DWORD Value dialog appears.




- Set the Value data, by typing one of the following options:

  0 - (zero) show no matches

  1 - show only user matches

  2 - show only group/alias matches

  3 - show groups/aliases and user matches

  The default behavior—without the Registry value `MailLookup` defined at a workstation—is 3.
- Click OK to save the setting and close the dialog box.

8. Close the Registry Editor window.

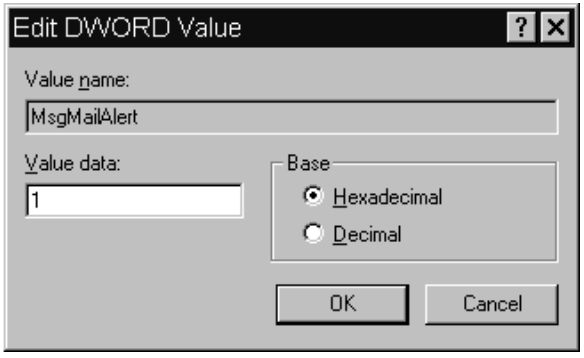
## MsgMailAlert

Enabling the Message Mail Alert feature allows you to change the alert behavior so that the iNEWS Workstation will flash message and/or mail alerts on the status bar for only 15 seconds, rather than persistently. By adding the `MsgMailAlert` variable, you can specify additional settings.

**To set the environment variable:**

1. Open Registry Editor. See “To access the Registry Editor:” on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **MsgMailAlert**.
6. Press Enter.
7. To set the Value data option:
   - Right-click on the `MesgMailAlert` value.
   - Choose Modify from the pop-up menu.

     The Edit DWORD Value dialog appears.




   - Set the Value data, by typing one of the following options:

     0 - disable - no alerts whatsoever on status bar

     1 - neither persistent - alerts flash for 15 seconds

     2 - only message alerts persistent

     3 - only mail alerts persistent

     4 - both alerts persistent - alerts will not go away until user has read all correspondence.

The default behavior—without the Registry value MsgMailAlert defined at a workstation—is 1.

8. Click OK to save the setting and close the dialog box.
9. Close the Registry Editor window.

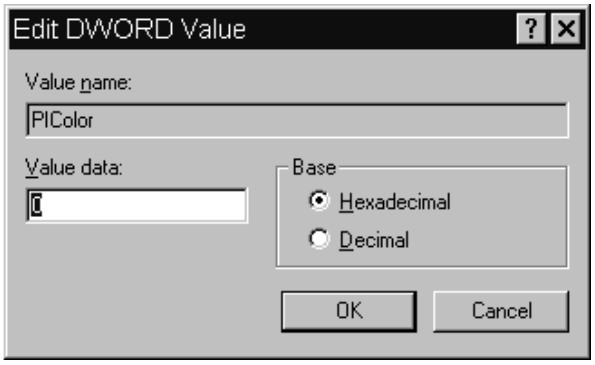
## PIColor

An individual iNEWS Workstation can have its presenter instructions text color changed via an environment variable called PIColor. If no environment variable exists, then the default color of red is used.

**To change presenter instructions text color:**

1. Open Registry Editor. See “To access the Registry Editor:” on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **PIColor.**
6. Press Enter.
7. To set the Value data option:
   - Right-click on the PIColor value.
   - Choose Modify from the pop-up menu.

     The Edit DWORD Value dialog box appears.




- Set the Value data using the following hexadecimal format:

  `0x00RRGGBB`

  where RR, GG, BB are two bytes for each color.

*The leftmost two bytes (`00`) are not used. Also, If the PIColor has its value set to zero (`0`), the closed captioning text is black because zero corresponds to the color Black.*

- Click OK to save the setting and close the dialog box.

8. Close the Registry Editor window.

## RGB Hexadecimal Color Chart

Avid's PIColor and CCColor environment variables require RGB Hexadecimal Color codes. The following are some basic colors, along with their corresponding hexidecimal code values:

Complete RGB Hexadecimal Color Charts, with various color shades, can be found on the Internet.

### RGB Hexadecimal Color Chart

| Color | Hex | Default |
|---|---|---|
| Black | 000000 | |
| Blue | 0000FF | |

| Color | Hex | Default |
|---|---|---|
| Brown | 330000 | |
| Green | 008800 | (Default color for CCColor) |
| Orange | FF6600 | |
| Pink | CC0099 | |
| Purple | 660099 | |
| Red | FF0000 | (Default color for PIColor) |
| White | FFFFFF | |
| Yellow | FFFF00 | |

## ShowTimingBar

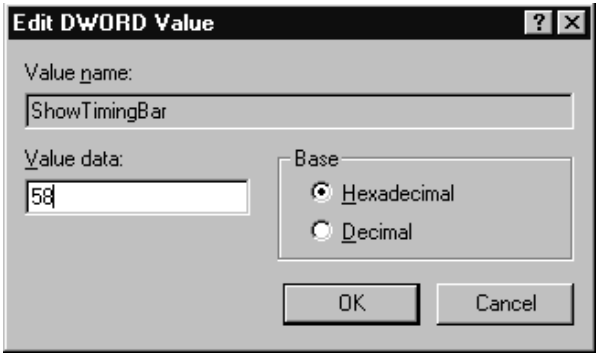
A system administrator can define which key on the keyboard is used to advance the timing bar during show timing. The default key is the space bar.

**To change the setting to a different key:**

1. Open Registry Editor. See “To access the Registry Editor:” on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **ShowTimingBar.**
6. Press Enter.
7. To set the Value data option:
   - Right-click on the `ShowTimingBar` value.

- Choose Modify from the pop-up menu.

  The Edit DWORD Value dialog box appears.




The `ShowTimingBar` Value data is determined by the Scan Code of the selected key on the keyboard. For instance, if the system administrator wants to use the F12 key to advance the timing bar, the Value data for the ShowTimingBar registry would be either the Hexadecimal code of `58` or Decimal code of `88`. See the following Scan Codes table for more information.

### Scan Codes

| Key | Decimal | Hexadecimal |
|---|---|---|
| ‘ ~ (accent/tilde) | 41 | 29 |
| 1 ! (exclamation point) | 2 | 02 |
| 2 @ (at symbol) | 3 | 03 |
| 3 # (pound sign) | 4 | 04 |
| 4 $ (dollar sign) | 5 | 05 |
| 5 % (percent) | 6 | 06 |
| 6 ^ (carrot) | 7 | 07 |
| 7 & (ampersand) | 8 | 08 |
| 8 * (asterisk) | 9 | 09 |
| 9 ( (open parenthesis) | 10 | 0A |
| 0 ) (close perenthesis) | 11 | 0B |
| - _ (dash/underscore) | 12 | 0C |

| Key | Decimal | Hexadecimal |
|---|---|---|
| = + (equal/plus) | 13 | 0D |
| Backspace | 14 | 0E |
| Tab | 15 | 0F |
| Q | 16 | 10 |
| W | 17 | 11 |
| E | 18 | 12 |
| R | 19 | 13 |
| T | 20 | 14 |
| Y | 21 | 15 |
| U | 22 | 16 |
| I | 23 | 17 |
| O | 24 | 18 |
| P | 25 | 19 |
| [ { (open bracket/brace) | 26 | 1A |
| ] } (close bracket/brace) | 27 | 1B |
| Caps Lock | 58 | 3A |
| A | 30 | 1E |
| S | 31 | 1F |
| D | 32 | 20 |
| F | 33 | 21 |
| G | 34 | 22 |
| H | 35 | 23 |
| J | 36 | 24 |
| K | 37 | 25 |
| L | 38 | 26 |
| ; : (semicolon/colon) | 39 | 27 |
| ‘ ” (accent/quote) | 40 | 28 |
| \ \| (backslash/pipe) | 43 | 2B |
| Left Shift | 42 | 2A |
| Z | 44 | 2C |
| X | 45 | 2D |

| Key | Decimal | Hexadecimal |
|---|---|---|
| C | 46 | 2E |
| V | 47 | 2F |
| B | 48 | 30 |
| N | 49 | 31 |
| M | 50 | 32 |
| , < (comma/less-than) | 51 | 33 |
| . > (period/greater-than) | 52 | 34 |
| / ? (slash/question mark) | 53 | 35 |
| Right Shift | 54 | 36 |
| CTRL (Control keys) | 29 | 1D |
| ALT (Alt keys) | 56 | 38 |
| Spacebar | 57 | 39 |
| ESC (Escape key) | 1 | 01 |
| F1 | 59 | 3B |
| F2 | 60 | 3C |
| F3 | 61 | 3D |
| F4 | 62 | 3E |
| F5 | 63 | 3F |
| F6 | 64 | 40 |
| F7 | 65 | 41 |
| F8 | 66 | 42 |
| F9 | 67 | 43 |
| F10 | 68 | 44 |
| F11 | 87 | 57 |
| F12 | 88 | 58 |
| INS (Insert key) | 82 | 52 |
| DEL (Delete key) | 83 | 53 |
| Home | 71 | 47 |
| End | 79 | 4F |
| Page Up | 73 | 49 |
| Page Down | 80 | 51 |

| Key | Decimal | Hexadecimal |
|---|---|---|
| Up Arrow | 72 | 48 |
| Down Arrow | 80 | 50 |
| Right Arrow | 77 | 4D |
| Left Arrow | 75 | 4B |
| NUM (Number Lock key) | 69 | 45 |
| / (divide on Numeric Keypad) | 53 | 35 |
| - (minus on Numeric Keypad) | 74 | 4A |
| + (plus on Numeric Keypad) | 78 | 4E |
| Print Screen | 55 | 37 |

## SyncToServer

The timing feature of iNEWS syncronizes the clock on the local workstation with the time set on the server when a user activates show timing. A user can use the Set Clock option from the Tools drop-down menu to manually override the clock synchronization. This feature is turned off by default, but a system administrator can turn it on at any workstation by creating a new registry value in the workstation.

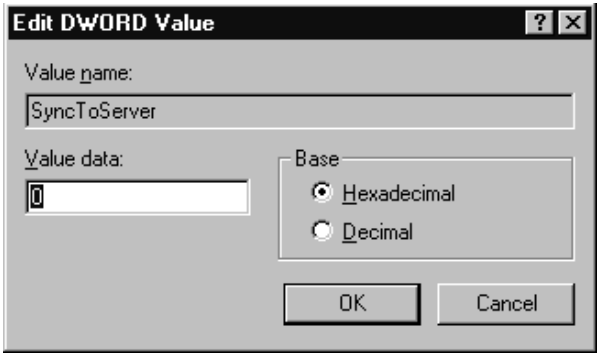
*The syncronized timing feature should be enabled only at those workstations used to time a show. If the* `SyncToServer` *registry value is not created and defined at a workstation, then the synchronized timing feature is disabled at that workstation.*

**This feature is not backwards compatible to iNEWS newsroom computer system software prior to version 1.2.4. Before enabling the synchronized timing feature by creating the `SyncToServer` Registry value in any workstation, the software for both iNEWS Workstation(s) and iNEWS Server(s) must be upgraded. iNEWS newsroom computer system will issue the following message if the server does not support the sync-to-server feature: Timing synchronization failed. Not supported on this server.**

**To enable the syncronized timing feature:**

1. Open Registry Editor. See “To access the Registry Editor:” on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **SyncToServer.**
6. Press Enter.
7. To set the Value data option:
   - Right-click on the SyncToServer value.
   - Choose Modify from the pop-up menu.

     The Edit DWORD Value dialog box appears.




   - Set the Value data. Type `0` (zero) to disable the synchronized timing feature, or `1` to enable it.
   - Click OK to save the setting and close the dialog box.
8. Close the Registry Editor window

# Delete_Notify

When a deleted queue record was received by the rxnet utility program (from a txnet utility program on another news system) into an update queue, and the queue had a notify group associated with it, an alert was sent to the notify group with the queue name and title (slug) of the deleted entry (story). This is no different than the notification for new or modified stories entering the queue. This resulted in users wasting time searching for deleted stories that were no longer in the queue.

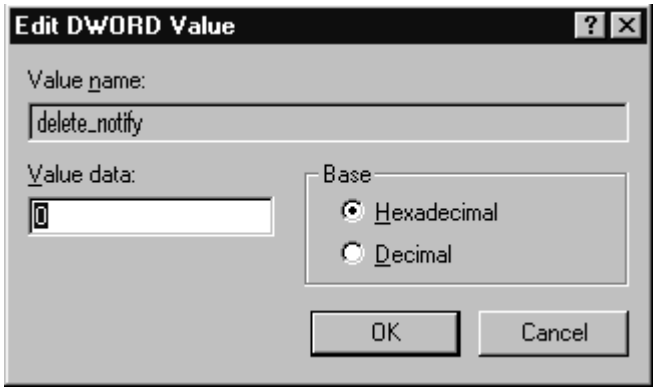
Action servers and txnet links behaved the same way as the rxnet utility program. The utility programs (servers) do not send out notifications when deleting queue entries.

As a precaution for sites which may want such notification to occur, an environment variable can be set to restore the former operation. The environment variable is defined in the workstation's Registry as `delete_notify`. It must be added in the Environment key of the Registry and should be set to a non-zero value.

**To change the setting for `delete_notify`:**

1. Open Registry Editor. See "To access the Registry Editor:" on page 274 for more information.
2. Navigate to the Environment key, and open it.
3. Right-click on the right side of the Registry Editor window. A pop-up menu appears.
4. Select the DWORD Value option to create and define a new registry value of type DWORD in the Registry Editor.
5. Type the name of the new value: **`delete_notify`**
6. Press Enter.
7. To set the Value data option:
   - Right-click on the **`delete_notify`** value.
   - Choose Modify from the pop-up menu.

The Edit DWORD Value dialog box appears.




- Set the Value data. Set the Environment key of the Registry to a non-zero value.
- Click OK to save the setting and close the dialog box.

8. Close the Registry Editor window

# Environmental Variables for Servers

You can control some of the behavior of rxnet by putting some enviroment variable settings in the `/site/env/nxserver file.`

*If the `/site/env/nxserver` file or its parent directory `/site/env` do not exist you need to create the directory using the `mkdir /site/env` command, before you can edit the file. Contact Avid Customer Support if you need assistance with this procedure.*

The specific variables are:

- RXDEBUG=<level>
- RXDEBUGFILE=<filename>
- RXSITELISTSZ=<size>
- RXSITEFORMAT=<format>
- RXSEPFIELDLIST=<filename>

- RXSITEIDLE=<seconds>
- RXOLDNLST=1

**RXDEBUG=<level>**

An rxdebug level of 2 produces the command traffic to/from rxnet.

**RXDEBUGFILE=<filename>**

The RXDEBUGFILE file allows you to cause rxnet diagnostics to be put into <filename>.<process id> instead of the console.

**RXSITELISTSZ=<size>**

The list size controls the number of stories included in a queue list when rxnet responds to an FTP list command. The default value for this is 300.

**RXSITEFORMAT=<format>**

The format controls which transfer protocol is initially used. This can be set to nsml or sep. The default is nsml. This is how you get a foreign application, such as CNN which operates in sep but did not use the FTP SITE command to change the format to sep.

**RXSEPFIELDLIST=<filename>**

RXSEPFIELDLIST defines a file name. That file is processed as a list of field substitutions. For instance,

RXSEPFIELDLIST=/site/env/substitutes

in the /site/env/nxserver file, rxnet inherits this variable when invoked.

The format of each line of the field substitution file is:

<field identifier> = <field name>

where <field identifier> is a single character SEP field

identifier and <field name> is a valid nsml field name. An example would be 1 = video-id. This causes the contents of the APP1 field, identified by 1, to be put into the nsml field video-id instead of the usual APP1-1.

**RXSITEIDLE=<seconds>**

The idle time value may be set from the client via SITE IDLE=<seconds> where <seconds> is a count of seconds. For instance:

SITE IDLE=900 sets the idle timeout to 15 minutes

SITE IDLE=0 disables the timeout

The client can query the rxnet server to find out what the current idle timeout is with `SITE IDLE`. You can also define an idle timeout value to override the localtimeout setting by inserting `RXSITEIDLE=<seconds>` in the file `/site/env/nxserver`.

*You can also define an idle timeout value to override the "localtimeout" setting by inserting `RXSITEIDLE=<seconds>` in the file `/site/env/nxserver`.*

**`RXOLDNLST=1`**

FTP clients that use the `MGET`command make an NLST request to get the list of story/file names for retrieval. The `LIST` and `NLST` commands produce the identical output, a directory listing. The `NLST` command produces a name list, that is a list of subdirectories in the directory and a list of stories id's in a queue.The iNEWS FTP Server produces a list of names without story attributes. If you want the iNEWS Server to include story attributes, put `RXOLDNLST=1` in the rxnet environment. If you do not want story attributes included, use `RXOLDNLST=0`.

*The bold entires are exactly what to place in the file. Do not use an export statement after the line.*

# Appendix G

# Managing Traits at the Console

Chapter 4 in the *iNEWS Newsroom Computer System Setup and Configuration Manual* explains how the system administrator can access and change the various user traits associated with each user's account from an iNEWS Workstation. In that same book, Chapter 5 explains how to manage database traits from a workstation, and Chapter 6 explains how to create groups and apply the system's group-related features to customize the system's security and usage from a workstation.

This Appendix shows how some of the information also can be viewed and modified from the console.

This appendix contains the following main sections:

- Viewing User Traits from the Console
- Modifying User Traits from the Console
- User Traits Console Command Summary
- Changing Database Traits from the Console
- Database Traits Console Command Summary
- Managing Group Traits at the Console

# Viewing User Traits from the Console

From the console, use the `list u-v` command to get user trait information. The console will display a verbose list of user accounts.

To get information about a single user, follow the command with the User ID of the specific user. For instance, if you wanted to access the user account, danielmi, the command would look like this:

```
list u-v danielmi
```

The result of the command will look something like this:

```
user      rr    kb   su   m    SOEKCVTH   sc   queues

danielmi  180   0    n    i    -OEKCVTH   sc   dest: PEOPLE.D.DANIELMI.NOTES
                                               home: PEOPLE.D.DANIELMI
                                               mail: PEOPLE.D.DANIELMI.MAIL


NRCS-A$
```

The first line of the display lists the traits; the other lines list specific values for the traits.

The following is the interpretation of the sample command results previuosly displayed:

- User `danielmi` has a read rate (rr) of 180.
- The keyboard preference (kb) is 0.
- The user is a superuser (su). The `n` means “news superuser.” A minus (-) would appear if the user is not a superuser.
- The edit mode (m) is insert.
- The traits indicated by SOEKCVTH and sc are explained in the “User Traits Summary” on page 305. See also “list u” on page 144.
- The users destination queue is `PEOPLE.D.DANIELMI.NOTES`.
- The users home directory is `PEOPLE.D.DANIELMI`.
- The users mail queue is `PEOPLE.D.DANIELMI.MAIL`.

“User Traits Console Command Summary” on page 304 shows user traits, their Console abbreviations, and detailed information about them.

# Modifying User Traits from the Console

You must be a superuser or user manager (`umanager`) to change user traits. For an explanation of the superuser privileges and for an explanation of the `umanager` account and privileges, see *iNEWS Newsroom Computer System Setup and Configuration Manual*.

Use the `utraits` command (which requires superuser privileges) to modify a user's traits from the console. The syntax of the `utraits` command is:

`utraits` *`userid`* {`trait` *`new-value`* [`trait` *`new-value`* `...`}| {`+`|`-`*`flag`*}

For instance, to set the read rate for a user named Daniel Mitchell, whose User ID is `danielmi`, the command would look something like this:

```
NRCS-A# utraits danielmi readrate 195
1 user records modified
NRCS-A#
To give him superuser privileges:
NRCS-A# utraits danielmi su n
1 user records modified
NRCS-A#
To take superuser privileges away from him:
NRCS-A# utraits danielmi su -
1 user records modified
NRCS-A#
```

The blacklist trait is a flag; that is, it is either on or off. You grant flag traits with a plus sign; you take them away with a minus sign. To blacklist the user:

```
NRCS-A# utraits danielmi +b
1 user records modified
NRCS-A#
To remove him from blacklist status:
NRCS-A# utraits danielmi -b
1 user records modified
NRCS-A#
```

You can change more than one trait at a time. For instance, to give this user keyboard 3 and make SHOW.RUNDOWN his destination queue, type:

```
NRCS-A# utraits danielmi key 3 dest show.rundown
1 user record modified
NRCS-A#
```

## Changing a Users' Password

**To change Smith's password to `changeme`:**

1. Log on as a superuser.
2. Type the `utraits` command using the following format:
   `utraits <username> password <newpassword>`

   For instance,

   ```
   NRCS-A# utraits smith password changeme
   1 user records modified
   NRCS-A#
   ```

**Since everything you type at the console is recorded in console history, consider using the force command to require the user to change their password the next time they log in. This will help prevent someone from using passwords obtained from the console history.**

For users who do not change their passwords, as instructed, use the `force` console command to require them to change passwords at their next login. The following is the force console command format:

`force  <user or group names>`

The most common way to use this command is to require a particular user or users to change their password the next time they log in.

For instance, suppose that you have been unable to convince Mitchell and Schofield to change their passwords. As a last resort, to require that they do so the next time they log in, log on as a superuser and type the following:

```
NRCS-A# force mitchell schofield
```

A message similar to the following appears:

```
Users who will be forced to make password changes on next login:
---------------------------------------------------------------
mitchell   schofield
2 users qualified out of a domain of 2 users, and were updated.
```

The `force` command tells you who is going to be required to change their passwords. The example above reports that it will make both Mitchell and Schofield change their passwords.

If you find that a user does not have a password, use the force command to require the user to select a password the next time he or she logs in.

The following example uses the `force` console command to make Weisman select a new password at the next login. To do this, type the

```
NRCS-A# force weisman
```

A screen similar to the following appears:

```
Users who will be forced to make password changes on next login:
---------------------------------------------------------------
weisman
1 users qualified out of a domain of 1 users, and were updated.
```

You can use the `force` console command to require that anyone who has not changed passwords since a certain date or within a certain date range do so. You can also use this command to force a particular group of users or all your users to change their passwords.

To do this, use the `force` console command with the following format:

```
force [-q] [passchg<date or date range>] <user or group names>
```

Normally, the `force` command tells you which users must change their passwords the next time they log in. If you would rather not see this display, suppress it with the `-q` option.

You may want to use the `force` command to require all users who last changed their passwords prior to a certain date to change their passwords the next time they log in. You can do this by specifying a date in the command, as shown in the previous format description.

The `force` command recognizes dates in the same way the `list u` console command does; you can specify relative dates, absolute dates, and date ranges. This command applies only to people who last changed their passwords within the date parameter you set.

If you specify a value for the date parameter, the `force` console command works only on those users who are among those you specify, and whose last password change falls within the criteria set by the date parameter.

For instance, the following command affects only those members of the producers group who last changed their password before July 5, 2004:

```
NRCS-A# force passchg\<05JUL2004 producers
```

To force all users to change their passwords, put a hyphen (`-`) in place of any user or group names. This is especially useful in combination with a date parameter. For instance, to force all users who last changed their passwords more than 90 days ago, become a superuser and type the following:

```
NRCS-A# force passchg\<90. -
```

A message similar to the following appears:

```
Users who will be forced to make password changes on next login:
----------------------------------------------------------------
erickson   mccormack   arlin
3 users qualified out of a domain of 62 users, and were updated.
```

## Listing Users Who Do Not Have Passwords

To check for users who do not have a password from the console, type:

```
list password= u
```

*Ensure that you include a space between the = and the u*

The following command lists every user who does not have a password.

```
NRCS-A: list password= u
user                rr   kb   su   mode  destination
weisman             0    0    -    o
```

In the previous example, there is one user, `weisman`, who does not have a password.

To find out who has not changed their password within a specified period of time, use this form of `list u`:

```
list  passchg<date>  u  [<domain>]
```

For instance, to list users who have not changed their password in the last 90 days, enter `list passchg` followed by `<` and the number of days you want to specify (90 in this case) and a period (`.`). Ending the number with a period indicates the value is in days; no period indicates hours. There must be no spaces between `passchg`, the `<`, and the number of days.

For information on users who have not changed their password within the last 90 days, type:

```
list passchg\<90. u
```

A screen similar to the following appears:

```
User    DEV   Date Created     Last Login       Last Password
------- ----  -------------    --------------   -------------
levy    427   02JAN2004 10:50am24JUL2004 9:03am 06JAN2004 9:50am
```

As you can see in the previous example, this produces a listing with:

- Name of user
- Device where the user logged in last
- When user became a user
- Date of the last login
- Date when password was last changed

In the previous example, only user Levy has not changed passwords within the last 90 days. So, use the `send` console command to send a message reminding Levy to change passwords:

```
NRCS-A: send levy Please change your password.
```

A message similar to the following appears:

```
message sent to levy
```

Use the less than (<) and greater than (>) operators to specify whether you want to list people who last changed their password *before* (<) or *after* (>) a certain date.

*When metacharacters— | & ; ( ) < > space or tab— or control operators—|| & && ; ;; ( ) | <newline> are used in a command string threy must be escaped with a backslash character.*

The following are some examples:

| | |
|---|---|
| `list passchg\<05JUL2004 u` | produces a list of all users who last changed their password before July 5, 2004 |
| `list passchg\>10. u` | produces a list of users who have changed their password after 10 days ago (that is, within the last 10 days). |

You can also use `list u` to list people who last changed their passwords sometime between two dates, such as between June 15, 2004 and July 1, 2004. You can even use this command to check a single user or a group of users.

To the afore mentioned, use this format of the `list` command:

```
list passchg\>date1\<date2 u [<user or group names>]
```

*The date1 and date2 parameters are not surrounded with greater than and less than characters here as is customary for parameters. These characters are also used in the command alone, and this could cause confusion.*

You must follow `passchg` with a date. This date may be a relative date, an absolute date, or a date range. Also, there must be no spaces between `passchg` and the date or date range, or the `list u` command does not work correctly.

A *relative date* is one that you specify as some time prior to the present date, as in `list passchg\<90. u`. In the previous example, we used a relative date to find out which users had last changed their password prior to

90 days ago. Remember, ending the number with a period (.) indicates that the value is in days; no period indicates hours.

An *absolute date* specifies an actual calendar date. In the following example, we use an absolute date to find out which users last changed their passwords before August 5, 2004, by typing the following:

**list passchg\<05AUG2004 u**

Information similar to the following appears:

```
User    DEV   Date Created     Last Login       Last Password
------- ----  -------------    ---------------  -------------
mitchell495   02JAN2004 10:55am24JUL2004 9:00am 07JAN2004 9:50am
```

You must specify absolute dates in DDMMMYYYY format. You must enter the days in double-digit format, meaning you must add a leading zero to single digit days, such as 05. Also, you must enter months as they are defined in the Words dictionary.

You can also specify a date range. This way, you can list users who changed their passwords sometime between two specific dates (date1 and date 2). For instance, to see if anyone changed their password after August 1, 2004, and before August 15, 2004, type the following:

```
list passchg\>01AUG2004\<15AUG2004 u
```

Information similar to the following appears:

```
User    DEV   Date Created     Last Login       Last Password
------- ----  -------------    ---------------  -------------
loyd    433   02JAN2004 11:50am25AUG2004 9:01am 05AUG2004 9:05am
```

You can also use list u to check on particular users or groups. To do this, follow the u in the command with user names or groups you want to check.

For instance, suppose that of all your users, only Mitchell and Schofield regularly forget to change their passwords. To see if they have not changed their passwords in the last 90 days, type the following:

**list passchg\<90. u mitchell schofield**

Information similar to the following appears:

```
User    DEV   Date Created     Last Login       Last Password
------- ----  -------------    --------------   -------------
mitchell495   02JAN2004 10:55am24JUL2004 9:00am 07JAN2004 9:50am
```

This causes `list u` to report only on Mitchell and Schofield, and only if they have not changed their passwords within the specified period of time. In the example above, Mitchell has not changed his password in the last 90 days, but Schofield has.

If you specify a group, such as producers, `list u` checks members of that group and then reports those that have not changed their passwords in the specified period of time.

Once you have set a policy on how often people must change their passwords, use `list u` regularly to ensure that no one forgets to do this within the prescribed period of time. If one or more users do not change their passwords often enough, use the `force` command to force them to do so. See "Changing a Users' Password" on page G-298 for more information.

# User Traits Console Command Summary

The following User Traits Summary table is a summary of iNEWS user traits. The first column shows the trait name as it appears in the Modify User Account dialog box, which is explained in detail in Chapter 4 of the *iNEWS Newsroom Computer System Setup and Configuration Manual*. The second column lists actual commands for assigning or removing each trait. The third column contains an explanation of the trait and example of utraits console command lines.

## User Traits Summary

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example |
|---|---|---|
| Superuser | `su n \| su -` | The user has superuser privileges. |
| Blacklisted | `+b \|-b` | The user cannot log in to the iNEWS system. |
| Simplified | `+s \|-s` | The user has limitations, as defined by the Simplified User Settings.<br>Examples:<br>**`utraits palmer su n`**<br>**`utraits palmer su -`**<br>**`utraits palmer +b`**<br>**`utraits palmer +s`** |
| Insert | `editmode i` | Everything the user enters in a story is inserted at the current cursor position, moving the following text over.<br>Example:<br>**`utraits palmer editmode i`** |
| Overwrite | `editmode o` | Everything the user enters in a story overwrites the character under the cursor. This is the default.<br>Example:<br>**`utraits hansen editmode o`** |
| Home | `home` | Sets the user's home directory, which usually contains the user's Mail and Notes queues.<br>Example:<br>**`utraits loyd home people.l.loyd`** |

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example |
|---|---|---|
| Destination | `destination` | Specifies the user's destination, which is usually a queue he or she uses frequently, such as the Notes queue.<br>Example:<br>**`utraits loyd destination people.l.loyd.notes`** |
| Mail | `mail` | Specifies the user's Mail queue, where any mail addressed to that user is placed.<br>Example:<br>**`utraits loyd mail people.loyd.mail`** |
| Read Rate | `readrate` | The user's spoken reading rate in words per minute.<br>The system uses the readrate of the designated presenter to determine the audio (air) time of a story.<br>Example:<br>**`utraits vandenberg readrate 180`** |
| User Name | `realname` | The user's real name (as contrasted with their account's User ID name). For instance, John Chapman may have a User ID of `jchapman`; his real name is John Chapman. |
| Video Browse | `+vb \| -vb` | Specifies user can search the video server.<br>Example:<br>**`utraits palmer +vb`** |

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example |
|---|---|---|
| Broadcast Control | `+td \| -td` | Enables the user to employ the technical director's workstation for broadcast control.<br>Example:<br>**`utraits miller +td`** |
| Connect Services | `+C \| -C` | Allows user to connect to any service defined in the system.<br>Example:<br>**`utraits smith +C`** |
| Toolbars | `+cs \| -cs` | Allows users to create custom toolbars.<br>Example:<br>**`utraits carver +cs`** |
| Color Highlights | `+cc \| -cc` | Allows users to configure custom status colors.<br>Example:<br>**`utraits bagley +cc`** |
| Highlight Read Stories | `+hr \| -hr` | Specifies that unread stories in the queue are highlighted on the user's screen.<br>Example:<br>**`utraits davis +hr`** |
| Reorder Stories | `+o \| -o` | Specifies that user can create and remove stories from a folder or queue.<br>Example:<br>**`utraits lumpkin +o`** |

| Name in Modify User Account dialog box | Utrait Console Command | Definition / Example |
|---|---|---|
| Create/Kill Folders/Queues | `+er \| -er` | Specifies that user can create and remove entire folders and queues.<br>Example:<br>**`utraits delany +er`** |
| Kill All Stories | `+ka \| -ka` | Specifies that user can select an entire queue and move stories in it to the Dead queue.<br>Example:<br>**`utraits richards +ka`** |
| Password button | `password` | Provides capability to change the user's password.<br>Example:<br>**`utraits jordan pass changeme`** |
| Force Change check box | `force` | Specifies that user will be forced to change his/her password the next time he or she logs in. This command line does not require `utraits` typed in front of it, as shown below.<br>Example:<br>**`force jordan`** |
| User Preferences button; Sessions tab; Keyboard field | `key ###` | Assigns a default keyboard (set of macros) to the user account.<br>Example:<br>**`utraits vandenberg key 048`** |

# Managing Database Traits from the Console

## Getting Basic Information

To obtain information about a particular queue or directory from the console, type list d, followed by the name of the directory or queue.

For instance, to obtain information about the WIRES.ALL queue, type:

```
list d wires.all
```

information similar to the following appears:

```
SRPlo-LIsUGQSXWFiT sortfield purge dis mbox directory
-R-----I---Q-X----  TITLE P4.0 D1          - WIRES.ALL
```

A trait listing begins with a header line containing the letters `SRPlo-LIsUGQSXWFiT`; each letter in the header line represents a particular database trait. That is, in the previous example, the second letter in the header line (an `R`) stands for the database trait Read-only.

When one of these traits is on, the letter representing that trait appears in the second line. For instance, the `R` in the second line of the example indicates the Read-only trait is on:

```
SRPlo-LIsUGQSXWFiT sortfield purge dis mbox directory
-R-----I---Q-X----  TITLE P4.0 D1          - WIRES.ALL
```

A hyphen in the second line indicates that trait identified above it is off. For instance, the first trait in the header, an `S`, represents a sequentially ordered directory or queue. Because the second line has a hyphen below the Sequential trait indicator (the `S`) means that `WIRES.ALL` is alphabetically, rather than sequentially, ordered.

Another trait indicated as "on" for `WIRES.ALL` is the one represented by the letter `I`. This trait is Inverted, which means this queue displays the most recent stories at the top.

```
SRPlo-LIsUGQSXWFiT sortfield purge dis mbox directory
-R-----I---Q-X---  TITLE P4.0 D1          - WIRES.ALL
```

## Getting Detailed Information

To obtain a more detailed information about a directory or queue, add the verbose command, such as list d-v followed by the queue or directory name.

For instance, to list information about the queue called `RUNDOWNS.5PM`, type:

```
list d-v rundowns.5pm
```

A screen similar to the following appears:

```
SRPlo-LIsUGQSXWFiT sortfield purge ap,al, as  dis mbox
RUNDOWNS.5PM:
QSRP-----U-Q------ TITLE   P0    A000,000,000 D1 220
     rg=castread wg=castwrite ng=-
queue form=MCS-RUNDOWN story form=MCS-RUNDOWN
```

The first letter (`Q`) in the line under the queue name indicates that `RUNDOWNS.5PM` is a queue.

The `ap` near the end of the header line stands for Abstract Printer. This trait indicates which printer your system uses to automatically print abstracts for this queue.

The `al` that follows `ap` stands for Abstract Lines. This trait controls whether or not abstracting is done, and, if it is done, how many lines of each story are printed.

# Changing Database Traits from the Console

To change a database trait from the console, you must use the `dbtraits` command. The general format is:

```
dbtraits pathname [only][option value][+mode][...]
[-mode][...]
```

Database traits come in two types: options and modes.

- Options accept a range of values, such as setting 18 hours for a queue's purge interval.

  ```
  dbtraits rundowns.5pm purge 18
  ```

- Modes are traits that are either assigned or not—that is, they are either turned on or off. A trait preceded by a plus (+) turns a mode on. A trait preceded by a minus (-) turns it off. The following is an example:

  ```
  dbtraits rundowns.5pm +p
  ```

You can change several traits at the same time. For instance, the following command changes the queue to read-only status and also assigns the wires story form and the wires queue panel form to it:

```
dbtraits wires +r sform wires qform wires
```

## Changing a Parent Directory Only

When you change a directory's traits at the console, `dbtraits` also applies your changes to any subdirectories or queues in that folder. You can restrict your changes to the parent directory by following the directory name with the word `only`.

*This is the opposite of the way traits are assigned using the iNEWS Workstation. See the iNEWS Newsroom Computer System Setup and Configuration Manual for more information on viewing database traits.*

For instance, to turn on the Scripts directory's Sequential trait (`+s`) without also turning it on for any of the Scripts subdirectories or queues, type:

```
dbtraits scripts only +s
```

# Database Traits Console Command Summary

The following Database Traits Summary table is a summary of iNEWS database traits, (`SRPlo-LIsUGQSXWFiT`) as seen and used in connection with the `dbtraits` command at the console. These traits may also appear in a GUI format at the iNEWS Workstation. For more on database traits, see the *iNEWS Newsroom Computer System Setup and Configuration Manual*.

### Database Traits Summary

| Name | Location in Display SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Sequential | `SRPlo-LIsUGQSXWFiT`<br>`S----------------` | `+s\|-s` | Lists directories or queues in the order in which they were created. (The default is alphabetical.)<br>To order the `RUNDOWN.5PM` sequentially:<br>**`dbtraits rundown.5pm +s`** |
| Read Access | `SRPlo-LIsUGQSXWFiT`<br>`-R---------------` | `+r\|-r` | Indicates whether or not stories in the queue are in read-only mode.<br>To set the `SHOW.5PM.SCRIPTS` queue to read-only mode, type:<br>**`dbtraits show.5pm.scripts +r`**<br>When this mode is turned off, users opening stories have them in edit-lock mode. This is the logical setting for any queue in which people will be changing stories.<br>Turn Read Access on for queues in which people are likely to read but not change the stories. |

| Name | Location in Display<br>SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Printable | SRPlo-LIsUGQSXWFiT<br>--p--------------- | +p\|-p | Indicates whether you can use the print command on all stories in the queue with a single command when systen printing.<br><br>To enable users to print all stories in the SHOW.5PM.SCRIPTS queue with a single command:<br>**dbtraits show.5pm.scripts +p**<br><br>This trait does not interfere with your use of the print story or print script commands on individual stories in the queue. |

| Name | Location in Display SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Queue Being Ordered | `SRPlo-LIsUGQSXWFiT`<br>`---lo----------` | `-o` | This is an indicator, rather than database traits. It indicates the queue's order status.<br><br>**`l`** indicates that the queue is currently order locked to prevent more than one user from reordering stories in a queue at the same time.<br><br>To find out who is ordering a queue, read the `Busy` error message you get when you try to order the queue. If no one is actually ordering the queue, then it has an invalid order lock on it.<br><br>**`o`** indicates the queue was once sorted, but has since been ordered. When a sorted queue is ordered, the system stops sorting stories as they enter the queue. To indicate that a sorted queue has been ordered, the system replaces `sorted` on the screen with `order`.<br><br>The dbtrait command `-o` can be used to remove the ordered attribute (indicator) and make the queue resume sorting from the console. There is no `+o` to apply the ordered attribute.<br><br>To find out who last ordered the `SHOW.5PM.SCRIPTS` queue:<br>**`list d-o show.5pm.scripts`** |
| Locked Queue | `SRPlo-LIsUGQSXWFiT`<br>`------L-----------` | | Not a database trait; indicates a user has locked the queue. Only a superuser or someone who knows the correct key for the lock can remove it.<br><br>To find out who last locked the `SHOW.5PM.SCRIPTS` queue:<br>**`list d-u show.5pm.scripts`** |

| Name | Location in Display<br>SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Inverted | SRPlo-LIsUGQSXWFiT<br>-------I---------- | +i\|-i | Indicates whether or not stories in a directory or queue will be displayed with the most recent one at the top.<br>To display stories in the WIRES.ALL queue, with the most recent one at the top:<br>**dbtraits wires.all +i** |
| Sorted | SRPlo-LIsUGQSXWFiT<br>--------s--------- | +so\|-so | Indicates whether or not the stories in a queue will be sorted by a form field you choose (usually the title field).<br>To turn on the sorted trait for the SHOW.5PM.SCRIPTS queue:<br>**dbtraits show.5pm.scripts +so** |
| Update | SRPlo-LIsUGQSXWFiT<br>---------U-------- | +u\|-u | Indicates whether or not stories in a queue will be replaced as new versions are moved or copied to it.<br>To indicate that stories in SHOW.5PM.SCRIPTS should be replaced as new versions are moved or copied to it:<br>**dbtraits show.5pm.scripts +u**<br>The update trait does not affect stories that are restored from tape. If you restore a story to a queue that already contains a version of that story, you will have two versions of the same story, even if the queue's update trait is on. |

| Name | Location in Display<br>SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| General | SRPlo-LIsUGQSXWFiT<br>----------G------- | +g\|-g | Indicates whether or not stories in a queue should retain their security restrictions when they are moved to another queue.<br>For instance, access to stories in the Dead queue would normally be unrestricted. To prevent people from opening restricted stories once they are moved to the Dead queue:<br>**dbtraits dead +g** |
| Queue Operations Allowed | SRPlo-LIsUGQSXWFiT<br>-----------Q------ | +q\|-q | Indicates whether or not the kill, move, or duplicate operations can be performed against an entire queue.<br>Setting the trait "off" for a particular queue does not affect the ability of people to kill, move, or duplicate individual stories in the queue, as long as they have appropriate permissions.<br>To allow users to kill, move, or duplicate the entire SHOW.5PM.SCRIPTS queue:<br>**dbtraits show.5pm.scripts +q** |

| Name | Location in Display SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Save Version | SRPlo-LIsUGQSXWFiT<br>----------S----- | save -<br>\|n\|o\|a | Determines how many old story versions are retained in each queue.<br>To save only the last version of the People directory (this is the default) use:<br>**dbtraits people save -**<br>To save no version use:<br>**dbtraits people save n**<br>To save only the original version use:<br>**dbtraits people save o**<br>To save all versions use:<br>**dbtraits people save a** |
| Skip | SRPlo-LIsUGQSXWFiT<br>----------X---- | +x\|-x | Indicates whether or not a directory or queue is left out of database backups.<br>To leave the Dead queue out of the database backup procedure:<br>**dbtraits dead +x** |
| Watch Append | SRPlo-LIsUGQSXWFiT<br>----------W--- | +w\|-w | Assigned to queues that receive data from wire services. It allows the queue to monitor for new stories sent by the wire service, appends them to the wire queue, and immediately displays them to users who have that wire queue open. While this trait can be applied to any queue in the iNEWS newsroom computer system, it is crucial that it be assigned to queues receiving data from wire services. For instance, the command to assign Watch Append to the WIRES.ALL queue is:<br>**dbtraits wires.all +w** |

| Name | Location in Display<br>SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Forms Allowed | `SRPlo-LIsUGQSXWFiT-`<br>`--------------F--` | `+f\|-f` | Must be assigned to all queues in the forms directory. The forms will not work without this database trait applied. Additionally, this trait can be assigned to any queue in the database, but is usually only assigned to other queues that receive stories from other systems via `rx/tx` and then build forms for those stories, as needed. For instance, the command to assign the forms allowed trait to the `SYSTEM.FORMS.R` queue is:<br>**`dbtraits system.forms.r +f`** |
| Index | `SRPlo-LIsUGQSXWFiT`<br>`----------------i-` | `+index\|`<br>`-index` | Assigned to queues and directories that you want to be indexed by the Fast Text Search (FTS) server. Allows for quicker searching of the queue or directory. For instance, the command to assign this trait to the `WIRES.ALL` queue is:<br>**`dbtraits wires.all +index`** |
| Sortfield<br>`SRPlo-LIsUGQSXWFiT sortfield`<br>`Q----------------- Title` | | `sortfield` | Indicates which form field a sorted queue is sorted by. See “Sortfield” on page 321 for more information.<br>When the sortfield is on, all the stories in the queue are sorted. When used on a large queue, the command prompt and control will not be returned until the entire queue is sorted. This may take a long time dependin on the size of the queue. |
| Purge Interval | | `purge` | Indicates the “age” stories in a queue can reach before they are purged. See “Purge Interval” on page 323 for more information. |

| Name | Location in Display SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| AbstractPrinting | | `ap\|as\|al` | Indicates whether or not a queue is an abstract printing queue. See “Abstract Printer” on page 325 for more information. |
| Display Lines/Refresh | | `dis /+\|-refresh` | Indicates how many lines of each story in the queue are displayed. See “The dis Column” on page 327 for more information. |
| Queue Form | | `qform` | The form used to display information in the Queue panel. Fields selected should be a sub-set of fields used in the story form. A field included in the queue form that does not actually exist in the story form cannot be written to in the Queue panel.<br>In this example the rundown form is assigned as the queue form to the `SHOWS.6PM.RUNDOWN` queue:<br>dbtraits show.6pm.rundown qform rundown |
| Story Form | | `sform` | Indicates the form assigned to be used in the Story Form panel (of the Story panel) to display form fields.<br>In this example the futures form is assigned as the story form to the `ASSIGNMENTS.TODAY` queue:<br>dbtraits assignments.today sform future |

| Name | Location in Display SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Change Form | | `cform` | Changes the story form assignment for previously existing stories within a queue, and<br>restamps them with a new form name. The following example would change the way stories in the 6 o'clock show would appear, reassigning them to be displayed using a newly designed rundown form.<br>Example:<br>dbtraits show.6pm.rundown sform rundown-new cform<br>For this to take effect, you need to log out and log back in again. |
| Strip Form | | `stripform` | Removes embedded form display information from stories. Forms allowed queues stamp the look of the story form into the story. Assigning a different story form to one of these queues and running `changeform` on the queue would not affect the look of stories with embedded forms. You would need to strip the embedded "look" out of the story so it would then appear using the form assigned to that queue. |
| Mailbox | `mbox` | `mail` | Indicates mailbox assigned to the queue. See “Mailbox” on page 326 for more information. |

| Name | Location in Display SRPlo-LIsUGQSXWFiT | Mode / Option Keyword | Definition / Example |
|---|---|---|---|
| Groups | `wg= rg= ng=` | wg\|rg\|ng | Assigns a write, read, or notify group to queue or directory. Here are a few examples:<br>**dbtraits show.5pm rg=castread**<br>**dbtraits show.5pm wg=producers**<br>**See "Managing Group Traits at the Console" on page 329 for more information.**<br>See "Groups" on page 329 for more information. |
| Text Timing Clocks | `SRPlo-LIsUGQSXWFiT`<br>`----------------T` | `+t\|-t` | Turns on the Text Timing Clocks, which appear in the Story Text panel. They are:<br>TTC - Time to cursor<br>BLK - Time of blocked (highlighted) text<br>EST - Estimated read time of entire story<br>To turn on these clocks in the `RUN-DOWN.5PM`:<br>**`dbtraits rundown.5pm +t`** |

## Sortfield

The format of the `sortfield` information is:

SRPlo-LIsUG-QSXWFiT **sortfield**

`Q---------------` **`TITLE`**

This trait shows by which form field a sorted queue is sorted. A letter representing the form field appears in this position in the trait listing. A hyphen (-) here means that no sort field has been set. The system automatically uses the title field as the sort field.

This trait also determines which form field the computer searches during fast finds. In addition, the cursor is placed on this form field when a user displays stories in a queue.

Do not confuse this trait with the sorted queue trait, which determines whether the queue is sorted at all.

### Changing a Queue's Sort Field

Perform this procedure at a time of low system usage, because the `dbsort` command can impair system performance while it is running.

#### To change a queue's sort field from the console:

1. Esure no users are in the queue.

   Changing a queue's sort field does not affect what is seen by users already in the queue. If they try to use a displayed out-of-date sort field, it could cause confusion.

2. Use `dbtraits` to set the queue's sort field.

   **`dbtraits <queue name> sortfield <field name>`**

   If the queue is not already sorted, include `+so` at the end of the command to turn on its sorted trait.

   For instance:

   d**`btraits rundowns.5pm sortfield page-number +so`**

Turn on the sorted option (**`+so`**) before allowing users to re-enter the queue.

### Starting the Queue Sort Function from the Console

You can use the console to restore a ordered queue to its original sorted state and restart sorting in one of two ways:

- Use the `dbsort` command
- Use the `dbtraits -o` command

**To restart sorting at a queue using the `dbsort` command:**

1. Type the `dbsort` command and the queue name you want to sort. For instance:
   **`dbsort rundowns.5pm`**
2. Press Enter. Something similar to the following appears:
   `Sorting: <RUNDOWNS.5PM>`

*If the queue has a write group assigned to it, you must be a console superuser to use `dbsort`. Otherwise, you will see something like this:`RUNDOWNS.5PM Write group producers NOT sorted.`*

**To restart sorting at a queue using the `dbtraits -o` command:**

1. Type the `dbtraits` command, the queue name you want to sort, and `-o`. For instance:
   **`dbtraits rundowns.5pm -o`**
2. Press Enter.

Using either the `dbsort` or the `dbtraits` command turns off the ordered attribute, allowing the queue to resume sorting, as indicated by the sort trait.

## Purge Interval

The format of the `purge` information is:

```
 SRPlo-LIsUGQSXWFiT sortfield    purge dis mbox directory
Q-RP----I---QSXW--- TITLE       P3.0    D1  -    WIRES.ALL
```

The `purge` column in the `list d` output displays the directory or queue's purge interval, preceded by `P`. The purge interval determines how old stories in a queue can get before they are purged. Every hour, your system removes any stories older than their queue's purge interval and places these stories in the Dead queue. This process frees up space in your database for new stories.

You can set a purge interval as days or hours, or a combination of both. You distinguish between hours and days by using a decimal point.

To enter hours, type the number of hours, such as `24`.

To enter the purge interval in days and hours, type the number of days, a period, and the number of hours, such as `2.12`.

To set the purge interval in days, type the number of days followed by a period (for instance, `90.`). A purge interval of zero means the queue is never purged.

To set a queue's purge interval, use:

```
dbtraits  <queue name>  purge  <purge interval value>
```

For instance, to give all queues in the Wires directory a purge interval of two days, type:

```
dbtraits wires purge 2.
```

To list all queues in the database with a purge interval, use:

```
list  purge=<purge interval>   d
```

To list all queues in a directory with a purge interval, use:

```
list  purge=<purge interval>  d  [<directory name>]
```

## Abstract Printing

Abstract printing is only for system printing.

This group of traits controls whether or not a queue is an abstract printing queue. When a story is added to or modified in an abstract printing queue, the system automatically prints a portion of the story. Using these traits, you can control how much of each story is printed and the print style used to print the story.

Abstract printing is controlled by the following traits:

- Abstract Printer
- Abstract Lines
- Abstract Style

To list a queue's abstract printing traits, type **`list d-a`** followed by the queue name. For instance, to list the abstract traits of `WIRES.STATE`, type:

```
list d-a wires.state
```

A display similar to the following appears:

```
 SRPlo-LIsUGQSXWFiT sortfield    ap,  al, as   dis directory
Q----------------- TITLE        A000,000,000  D1  WIRES.STATE
```

The three abstract printing traits appear toward the middle of the listing under the headings `ap`, `al`, and `as`. There is no abstract printing facility for local printing.

### Abstract Printer

The abstract printer trait indicates which system printer your system uses to automatically print abstracts for this queue. There is no abstract printing facility for local printing. To select a printer, set this trait to the printer number you want to use. (This is the same number you would use to select that printer in a `print` command.) Use this form of `dbtraits`:

```
dbtraits  <queue name>  ap  <printer number>
```

For instance, to set `WIRES.STATE` to print abstracts on printer 1, type:

```
dbtraits wires.state ap 1
```

The format of the `abstract printer` information is:

```
 SRPlo-LIsUGQSXWFiT sortfield    ap,  al, as   dis directory
Q----------------- TITLE        A000,000,000  D1  WIRES.STATE
```

To list all queues in the database that have a particular abstract printer value, use:

```
list  ap=<printer number> d
```

### Abstract Lines

The abstract lines trait controls whether or not abstracting is done and, if it is done, how many lines of each story are printed.

Both `ap` and `al` must be non-zero for abstract printing to be enabled.

To set this trait, use:

```
dbtraits  <queue name>  al  <abstract lines value>
```

For instance, to set `WIRES.STATE` to print 10-line abstracts, type:

```
dbtraits wires.states al 10
```

If you do not want the queue to be an abstract printing queue, set this trait to zero. Otherwise, set this trait to the number of lines you want printed for each story. You can select a number of lines between 1 and 125. A value of 126 prints the entire story in script form. Setting abstract lines to 127 prints the entire story in story form.

Zero is the default value for this trait, which means the queue is not set up to print abstracts.

The format of the `abstract lines` information is:

```
 SRPlo-LIsUGQSXWFiT sortfield    ap,  al, as   dis directory
Q----------------- TITLE        A000,000,000  D1  WIRES.STATE
```

### Abstract Style

The abstract style trait indicates the print style your system uses when it prints an abstract from this queue. To choose a style, set this trait to the number representing the style you want. To list all queues in the database that have a particular abstract style value, use:

```
list  as=<style number>  d
```

To set this trait, use:

```
dbtraits  <queue name>  as  <abstract style value>
```

For instance, to set `WIRES.STATE` to print abstracts with style 3, type:

```
dbtraits wires.states as 3
```

You can select any style defined in `SYSTEM.STYLES`.

The format of the `abstract style` information is:

```
 SRPlo-LIsUGQSXWFiT sortfield    ap,  al, as   dis directory
Q----------------- TITLE        A000,000,000  D1  WIRES.STATE
```

## Mailbox

If a mailbox is assigned to a queue, the number representing that mailbox appears in the `mbox` column of the `list d` display.

The format of the `mailbox` information is:

```
 SRPlo-LIsUGQSXWFiT sortfield    purge dis mbox directory
Q-RP----I---QSXW--- TITLE        P3.0    D1  -    WIRES.ALL
```

To list all queues in the database with a particular mailbox, use:

```
list  mail=<mailbox number>  d
```

To assign a mailbox to a queue, use:

```
dbtraits  <queue name>  mail  <mailbox number> | <reserved
mailbox name>
```

For instance to assign mailbox 103 to the queue `WIRES.KEYWORDS.HOCKEY`, type:

```
dbtraits wires.keywords.hockey mail 103
```

## The dis Column

The `dis` column of the `list d` output represents two traits in one column: the preview lines trait (formerly known as display lines) and the refresh trait.

The format of the `preview lines` information at the console is:

```
 SRPlo-LIsUGQSXWFiT sortfield    purge dis mbox directory
Q-RP----I---QSXW--- TITLE       P3.0    D1  -    WIRES.ALL
```

The preview lines trait applies to lines previewed in the Queue panel of the iNEWS Workstation. The value of a queue's preview lines trait controls how many lines of each story in the queue are displayed. The number in the `dis` column indicates the preview lines setting. You can have a queue display as few as one line of each story or as many as 23 lines.

The second trait, refresh, is also listed under `dis`. Refresh controls whether the system automatically updates your screen if you view a queue someone else is changing. If the number under `dis` begins with a `D`, the queue is not refreshed automatically; if it begins with an `R`, it is refreshed automatically.

### Preview Lines

To change the value of a queue's preview lines trait, use:

```
dbtraits  <queue name>  dis  <number of lines>
```

Because screens can display a maximum of 22 lines of a story at a time, that is the maximum value used with this trait. The minimum value is `0`, which would show no preview lines.

For instance, to give `SHOW.RUNDOWNS` a one-line display, type:

```
dbtraits show.rundown dis 1
```

`SHOW.RUNDOWNS` now has a one-line-per-story display.

In a one-line display, the first line on the screen is the top line of the form assigned to that queue. The next 22 lines in the queue display the second line of each story's form. Queues that display more than one line per story begin with the top line of each story's form.

Choosing a value for a queue's display lines trait depends on which is more important: displaying as many stories as possible at one time, or showing a large portion of each story.

For instance, in a rundown queue it is important to display as many stories on the screen at a time as possible, so the display lines trait is usually set to one line per story.

In queues where people often browse (such as those in the Wires directory), display a small part of each story.

Stories in a queue display are usually separated by a blank line. The only exception is any queue that displays one line per story, such as a rundown queue. Stories in these queues are not separated from each other.

Type the `display` command to temporarily change the number of lines displayed per story for a queue you are viewing. This change applies only to a single user′s workstation and lasts only while the queue is being viewed.

### Refresh

To set the queue refresh trait at the console, use:

```
dbtraits  <queue name>  [+refresh | -refresh]
```

To turn on this trait for a queue, use `+refresh`; to turn it off, use `-refresh`. For instance, to turn on the refresh trait in `RUNDOWNS.5PM`, type:

**`dbtraits rundowns.5pm +refresh`**

Queues with refresh turned on display `R` in the preview lines column of `list d` output.

Use this trait only on important queues, like rundowns, that more than one person may modify simultaneously. This allows everyone working on that rundown to immediately see changes made by others.

*To automatically refresh a queue, your system must spend a long time monitoring workstations where users are viewing that queue. Using the refresh option on too many queues simultaneously greatly increases the amount of work your system has to do and may severely degrade its overall performance.*

# Managing Group Traits at the Console

## Groups

There are three primary group traits used to apply groups to queues and directories in the database: Read, Write, and Notify.

The `list d-v` console command lists each queue's assigned read, write, and notification groups. These groups restrict who can read or write stories in the queue and indicate who is notified when stories change in it. Each group is explained individually in the following sections.

To list group information for a queue at the console, use:

```
list  d-g  <queue name>
```

To list all queues in the database with a particular group assigned as their read, write, or notification group at the console, use:

```
list  rwng=<group name>  d
```

### Read Group

To set a queue's read group, use:

```
dbtraits  <queue name>  rg  <group name>
```

For instance, to assign a read group of producers to the queue `SHOW.5PM`, type:

```
dbtraits show.5pm rg producers
```

To list all queues in the database with a particular group assigned as their read group, use:

```
list  rg=<group name>  d
```

### Write Group

To set a queue's write group, use:

```
dbtraits  <queue name>  wg  <group name>
```

For instance, to assign a write group of producers to the queue `SHOW.5PM`, type:

```
dbtraits show.5pm wg producers
```

To list all queues in the database with a particular group assigned as their write group, use:

```
list  wg=<group name>  d
```

To list all queues in the database with a particular group assigned as their read or write group, use:

```
list  rwg=<group name>  d
```

### Notify Group

To set a queue's notification group, use:

```
dbtraits  <queue name>  ng  <group name>
```

For instance, to assign a notification group of producers to the queue `WIRES.WAR`, type:

```
dbtraits wires.war ng producers
```

To list all queues in the database that have a particular group assigned as their notification group, use:

```
list  ng=<group name>  d
```

## Restricting Access Using Read and Write Limitations

In addition to restricting access to various queues, you can use group access and usage restrictions to hide queues or directories by placing a strict read restriction on them.

For instance, the System directory is usually restricted so that only superusers can write stories there. You can hide this directory so it does not appear in the main directory for normal users. Set its read group to a group with no users, such as sysop. (Because system administrators can read everything in the database, they can see the directory.)

**To set the System directory's read group to `sysop` at the console:**

1. Log on to the console as a superuser so that the prompt appears like this:

   ```
   NRCS-A#
   ```

2. Type:

   ```
   NRCS-A# dbtraits system rg sysop
   ```

3. To apply changes you made to stories already in the directory and as a console superuser and type:

   ```
   NRCS-A# gtraits changegroup system
   ```

   A message similar to the following appears:

```
SYSTEM.CABLES  5 out of 5 changed
...
SYSTEM.WIRES 4 out of 4 changed
```

Now, only system administrators, logged in with an superuser account, can see the System directory on their screens.

You may also have a queue to which you want people to move or duplicate stories, but that you do not want anyone to read or go to. An example of this would be a confidential suggestions queue to which

people would move stories containing suggestions. Once placed in this queue, suggestions would be visible only to those authorized or to superusers.

For instance, to set the read group of the Suggestions queue to mngmt, as a console superuser, type:

```
NRCS-A# dbtraits suggestions rg mngmt
```

*All users you want to allow the capability to send suggestions need to have write access to the queue.*

To make changes take effect on stories already in the Suggestions queue, remain a console superuser and type:

```
NRCS-A# gtraits changegroup suggestions
```

Unless you are a member of the mngmt group, you can not see this queue in the Directory panel or open this queue. You may copy or move stories containing suggestions to this queue.

## Removing Directory or Queue Restrictions

To remove a directory or queue's read or write restrictions at the console, you first must log on as a console superuser—that is, the prompt should appear with a number sign (#) not a colon (:). To remove a restriction, type **dbtraits** followed by the name of the directory or queue you want to change, then **rg**, **wg**, **ng**, and a dash (**-**).

For instance, to remove both the read and write restrictions from the Phonelists queue:

As a console superuser, type:

```
NRCS-A# dbtraits phonelists rg - wg -
```

To apply these read and write group changes to stories already in the Phonelists queue, as a superuser, type:

```
NRCS-A # gtraits changegroup phonelists
```

Now, anyone can read and write stories in the Phonelists queue.

# Appendix H

# The Line Editor, `ed`

This appendix provides basic information about the line editor, called `ed`, including procedures for using it to make changes to important system files. These include the system configuration, system profile, and printer or wire profile files.

*Additional information may be obtained by typeing* **`info ed`** *at the console.*

This appendix contains:

- Launching ed
- Editing Commands
- Saving Changes
- Quitting ed

# Launching `ed`

When you modify a file, make the same changes to each server's copy of the file, or your system will not run properly. Select all servers before launching `ed` and opening a file for editing to ensure changes you make are applied to each server's copy of the file.

**To launch `ed` from the console:**

- Type **`ed`** followed by the file path and name to be edited. For instance, to edit a copy of the configuration file in the Site directory, you would type:

  ```
  NRCS-A$ ed /site/config.test
  3624
  ```

The line editor, `ed`, returns a number indicating the file size expressed as the number of characters, including spaces and returns. Also, the console prompt, `NRCS-A$`, changes to no prompt when you launch the editor.

If the file name specified does not exist or is a non-text file unsuitable for editing with `ed`, `ed` returns a question mark (?) followed by the file name. This is one way to create a new text file. For instance, a new text file called `newfile` is created when the following is typed:

```
NRCS-A$ ed newfile
?newfile
```

**Do not attempt to edit a non-text file such as a binary file. Doing so could cause undesirable results.**

## Specifying Lines to Edit

The line you are on presently is called the current line.

Because `ed` is a line editor, you navigate through the file by line numbers. For instance, when you open a file for editing, `ed` considers the last line in the file the current line. If you want to view or edit a different line, you must go to that line.

For instance, you can move to the first line of the file by typing 1 and pressing Enter. To move ahead five lines, you could type **`+5`**. To move back three lines, you could type **`-3`**.

*The line editor, ed, will respond with a question mark (?) if you try to move beyond the last line of the file. Additionally, you cannot type a minus (-) value greater than or equal to the current line number, because you cannot move to a line preceding the first line in the file.*

Within ed, pressing Enter with no line number reference or command will cause ed to make the next line in the file the current line, displaying that line as it goes. For instance, in the following example, the user selects line 19 in the file, then presses the Enter key three times. The line editor, ed, responds each time by displaying lines 20, 21, and 22, respectively.

```
19
;printer     11     2400          11                -      ;
Enter
terminal     12     19200         0        news     -      ;
Enter
terminal     13     9600-8n       1        news     -      ;
Enter
wire         14     2400          anpa7    AX       -      ;
```

When editing, it is necessary to specify the line number(s) to be acted upon. This can be done in several ways:

- Type the line number.
- Type starting and ending line numbers separated by a comma.
- Type period (.) to specify current line number.
- Type a dollar sign ($) to specify last line in the file.

The editing command to act upon the specified line(s) is typed immediately following the specified line(s). There should be no spaces. Here are some examples using the Print command, p.

| | |
|---|---|
| To make line 18 the current line, type: | **18** |
| To print (to screen) line 10 of the file, type: | **10p** |
| To print (to screen) lines 10-20, type: | **10,20p** |
| To print all lines from the current line to line 20, type: | **.,20p** |
| To print all lines in the file from line 80 to the end, type: | **80,$p** |
| To display the current line number, type: | **.=** |
| To display the line number of the last line, type: | **$=** |

To make the fifth line from the bottom current, type: **`$-5`**

## Searching the File

When you do not know the line number, but you want to locate a line containing a specific word, phrase, or number, you can use the search option. The line editor, `ed`, will search the file, starting at the current line, and display the line with the next occurrence of the specified text.

### To search for text:

1. Type a forward slash (/). Do not press the spacebar.
2. Type the text you want to locate, followed by another forward slash.
3. Press Enter.

For instance, if you want to find websession in the configuration file, type:

```
/websession/
;websession    900
```

In the previous example, `ed` found the word, websession, on line 900. If you want to repeat the search to locate further occurrences of that word, type a forward slash and press Enter again. For instance:

```
/websession/
;websession    900
/
;websession    901
/
;websession    903
```

In the previous example, the user repeated the original search command two more times. Each time, ed responded with the word searched and the line number where the next occurrence of the word appears. In each case, the current line becomes the line number displayed by `ed`.

### Searching Tips

Here are a few more tips for searching with `ed`.

- Remember to use spaces before and/or after text to further define your search string. For instance, type **`/ 25 /`** instead of **`/25/`** to avoid finding other numbers that contain the number 25, such as in the line:
  `net  n125  20`
- Remember that searches are case sensitive. For instance, searching for **`/PCU/`** does not find pcu.
- Remember that searches distinguish between spaces and tabs. In other words, if you use spaces and the file contains the information separated by a tab, you will not find it. For instance, the search example below will not work:

  You type: **`/pcu 20/`** (pcu and 20 are separated by a space)

  The line is: `pcu  20` (pcu and 20 are separated by a tab)

## Editing Commands

There are several basic editing commands you can use in ed to view, change, add, move, and delete text in a system file.

*Some commands, such as Add, Delete, and Insert, change the current line, while others do not. For instance, the Print command sets the current line to the number of the last line printed.*

Here is a list of editing commands, along with examples of their use:

| Command | Description & Examples |
|---|---|
| a | The Append command inserts one or more lines *after* the selected line. For instance:<br>**/websession/**<br>;websession 900<br>**a**<br>**inws 511 - 1 gnews - ;**<br>**inws**<br>**.**<br>In the above example, the user searches for websession, and ed responds by displaying the first line found containing that word. The user types **a**, presses Enter to start the append operation, and types information to be inserted in the file *after* the displayed line. The user then types a period (**.**) on a line by itself, which is very important because it terminates the append operation. Without it, successive lines typed by the user would be added to the file. |
| c | The Change command replaces the entire contents of the line addressed. For instance:<br>**21**<br>terminal 13 9600-8n 1 news - ;<br>**c**<br>**; not used**<br>**.**<br>In the above example, the user selects line 21, and ed responds by displaying line 21. The user types **c** on one line to start the change operation. On the second line, the user types replacement text and presses Enter, followed by a period on a line by itself. The period (**.**) is very important because it terminates the change operation. Without it, successive lines typed by the user would be added to the file. |
| d | The Delete command is used to delete the line(s) specified. For instance:<br>**27d** - Deletes line 27.<br>**30,35d** - Deletes lines 30 through 35.<br>**40,$d** - Deletes every line from 40 to the end of the file.<br>*After each Delete command, as well as after the Append, Change, and Insert commands, the file's lines are renumbered. For instance, if you delete line 27, then delete line 30, you actually delete the lines originally numbered 27 and 31 prior to the first Delete command.* |

| Command | Description & Examples (Continued) |
|---|---|
| g | The Global command allows the user to apply an editing command to all lines in the file that contain a specific word, phrase, or number. For instance:<br>**g/asws/s/asws/inws** - Finds all occurrences of asws and uses the Substitute command to replace the first occurrence of asws on each line found with inws. To replace all occurences on a line, add **/g** to the end of the command, such as: **g/asws/s/asws/inws/g**<br>**g/websession/d** - Finds all occurrences of websession and deletes the lines containing it.<br>Each line affected is displayed after the editing command is applied. |
| i | The Insert command inserts one or more lines *before* the selected line. For instance:<br>**/websession/**<br>;websession 900<br>**i**<br>**inws 511 - 1 gnews - ;**<br>**inws**<br>**.**<br>In the above example, the user searches for websession, and ed responds by displaying the first line found containing that word. The user types **i** and presses Enter to start the insert operation. The user then types two lines of information to be inserted in the file *before* the displayed line. On the last line, the user types a period (**.**) on a line by itself, which is very important because it terminates the insert operation. Without it, successive lines typed by the user would be added to the file. |
| m | The Move command removes the line(s) specified from their original location and inserts the line(s) *after* the target location. For instance:<br>**18m20** - Line 18 becomes line 20. Lines 19 and 20 become 18 and 19.<br>**1,5m$** - Moves lines 1 through 5 to the end of the file. |

| Command | Description & Examples (Continued) |
|---|---|
| `p` | Contrary to the name, the Print command does not send information to a printer. Instead, it *prints* text to the console screen. It is handy for viewing specific lines within a file. Typed alone with no line number references, the Print command displays the current line.<br><br>For instance, to print (to screen) lines 10-20, type: **`10,20p`**.<br><br>More examples of the Print command can be found in "Specifying Lines to Edit" on page 334. |
| `s/<old text>/<new text>` | The Substitute command substitutes a specific portion of a line with new text.<br><br>For instance:<br>**`3`**<br>`;AvidNews  Starter config  09JAN00`<br>**`s/AvidNews/iNEWS`**<br>`;iNEWS  Starter config  09JAN00`<br><br>In the above example, the user selects line 3, and `ed` responds by displaying line 3. The user substitutes the word `AvidNews` for `iNEWS` but does not alter anything else on the line. The line editor, `ed`, confirms the substitution by redisplaying line 3, incorporating the change. |
| `t` | The Copy command copies the line specified, and inserts a copy *after* the target location. For instance:<br><br>**`5t10`** - Inserts a copy of line 5 *below* line 10. The copy becomes line 11. The original line 11, if any, becomes line 12, and so forth.<br><br>**`20,30t50`** - Inserts copies of lines 20 through 30 *after* line 50. |
| `u` | The Undo command is used to cancel the effects of the last editing command entered. For instance:<br>`1,5m$`<br>`u`<br><br>In the above example, the user issues a command to move lines 1 through 5 to the end of the file, then types **`u`** to undo that command. |

## Saving Changes

The changes you make to a file are not saved immediately. This means you could quit (or exit) `ed` without saving changes if necessary. You must use the Write command to save modifications.

To save changes, type **w** and press Enter. The line editor, ed, will respond by displaying the file size, such as:

```
w
3624
```

*The Write command is case-sensitive. If uppercase* W *is used,* ed *will append the modified version of the file to the end of the original file version. This can quickly increase the file size and result in redundancy. Always use the lowercase* w.

## Quitting ed

To exit the line editor, type **q** and press Enter. For instance:

```
w
3624
q
NRCS-A$
```

In the previous example, the user saved changes first by using the Write command (**w**), then typed **q** to quit ed and return to a normal console prompt.

However, if you wanted to quit ed without saving your changes, you can do so. In this case, you would have to type the Quit command twice: the first time to notify ed you want to quit, and the second time to confirm that you want to quit without saving changes.

For instance:

```
q
?
q
NRCS-A$
```

In the previous example, ed responds to the first Quit command with a question mark (?) to remind the user changes were made to the file and not saved. This is a precautionary warning to help prevent a user from exiting ed and inadvertently losing changes that were not saved. When the user replies by typing the Quit command a second time, ed exits, abandoning any changes made.

# Index

## Symbols

/site/config 23

## Numerics

7-bit character set 255

## A

absolute date, defined 303
abstract printing
    abstract lines database trait 325
account created option 36
adding
    network services 23
    services 23
ASCII tables 255
assigning
    mailboxes to a queue at console 327
at command 48
at job list command 163

## B

backing up software 75
backup
    appending to tape 61
    continuation tape 63
    data to tape 57–??, 73–75
    dbdump command 57
    listing tape backup dates 66
    maximum stories found 71
    new tape 61
    procedures 56
    queues 60
    retrieve stories from tape 128
    searching
        a tape 68
        by word and date 70
        for stories, word and day 70
        for stories, word and month 71
    site files to tape 76
    sitedump command 76, 156
    system 156
    tape 123
    verifying 59, 62
bad story, fix link counts 133
blocks
    database organization 43
    locked 104
bottom console control command 116
bpoll command 163
broadcast command 120
    message 120
broadcast message 120
bscan command 164
building dialogs 20
busy stories 99

## C

capture dialog command 170

caution
database corruption 83
diskcopy 93
servers disconnect 81
cccolor 276
ccucmds 196
ccureset console command 152
changing
user passwords 31
character mapping
dbrestore 63
devices 173
service 173
checking for a locked block 105
cleaning the database
monthly 50–53
offline 50–53
commands
at 48
Avid personnel only 114
capture connect 20
date 49
dblines 48
dbserver 43
dialog 169
doc 49, 136
editing 337
grep 49
hogs 45
communication, testing 152
computer commands 116
configuration file
checking for errors 120
incorporating changes 120
configure command 120
connect
changing escape character 171
installing a service 23
mapping characters 173
connect command 121
waiting for 21
connecting a service 23
connecting devices
network resources 23
connectivity, testing with ping 107
console
managing groups 329
PRINTER display 117
printing history 117
console command
broadcast 120
ccureset 152
configure 120
connect 121, 153
dbclean 122
dbclose 122
dbdev 123
dbdump 123
dbfree 124
dblines 126
dboriginal 126
dbpurge 127
dbrestore 128
dbserver 130
dbsort 131
dbvisit 133
dial 134
diskclear 135
diskcopy 136
doc 136
edit 136
force 33, 35, 137, 298
grpcheck 137
gtraits 137
help 138
hogs 138
idiff 139
list a* 146
list B 142
list c 142

list d 142
list g 143
list s 144
list u 144, 304
logout 146
makeccutab 147
maketab 149
msgclean 150
offline 150
online 150
otod 151
print 152
reconnect 153
rename 154
restart 154
searchtape 155
send 155
shutdown 156
sitedump 156
siterestore 156
softdump 157
softrestore 157
startup 158
stop 159
su 159
utraits 160
version 161
wholockedit 162
wiredump 162
copying stories 44


## D


DAT tapes 56
database
blocks 43
checking for errors 48
cleaning monthly 50–53
cleaning offline 50–53
cleanup 50
error check 133
export stories if disconnect occurs 81
free up space 323
purge interval 42
purging cycle 42
retrieve stories from tape 128
size 124
understanding storage 43
database cycle 42
database maintenance 48
dbvisit 133
regular repair 133
database storage
create space by purging 323
increasing 46
maintaining 48–53
tracking 44
units 43
database traits
abstract lines 325
display lines 327
display lines for rundown queue 328
purge interval 323
skip flag 58
sortfield 321
date command 49
date1 and date2 302
dbclean console command 122
dbclose console command 122
dbdev console command 123
dbdump command 57, 61
dbfree 72, 124
space usage 46
dblines 126
explained 48
using log file 48
dboriginal 126
dbpurge 42, 127
dbrestore 64, 128
character mapping 63

tdv command 59
dbserver 43
space usage 46
syntax 130
dbsort 131
dbvisit 133
checking the database 52
cleaning database 50
Dead queue 42
introduction 42
reclaim space 130
reclaiming 46
retrieving a story 42
unbusying 100
defining messages and commands using dictionaries 196
delay dialog command 170
delete_notify 290
deleting runaway print jobs 104
destinationorder 278
devices
character mapping 173
diag dialog command 171
dial command 134
dialog commands
capture 170
delay 170
diag 171
echo 171
escape 171
expect 172
heol 172
introduction 169
list 169
map 173
mapin 173
mapout 173
message 174
pass 174
pause 174
stop 175
timer 175
type 175
wait 176
dialog for service 25
dialogs 20–22
building 20
capture command 20
creating a queue 20
dictionaries
case-shifting 240
ccumsgs 213
connect dictionary 230
defining messages and commands 196
dial dictionary 236
editing 198
incorporating changes 200
keyboard macros 237
mcs dictionary 242
printer messages dictionary 239
queues dictionary 223
restoring standard translations 201
space limitations 197
standard names 197
table space exceeded 201
telex dictionary 234
words dictionary 226
dictionary defaults, restoring 201
disaster recovery 73–75
disconnect
causes of 83
detecting 81
exporting stories 81
overview 80
recovery 84
types of 82
disk space, distribution and purging 42
diskclear command 135
diskcopy 136
caution 93

disks in use, list 123
display line
  rundown queue 328
  setting 327
distribution job list command 164
doc 49, 136
down keyword console control command 116
down number console control command 116
dup job list command 164

## E

echo dialog command 171
edit command 136
edit lock, unbusying 100
editing the dictionary file 198
environment variables 273
  cccolor 276
  delete_notify 290
  destinationorder 278
  msgmailalert 280
  nxserver 291
  picolor 282
  RXDEBUG= 293
  RXDEBUGFILE 293
  RXSITELISTSZ= 293
  server 291
  showtimingbar 284
  synctoserver 288
error checking, story 126
errors in the database 48
escape dialog command 171
every job list command 164
expect dialog command 172

## F

fast text search (FTS) 109–111
fgrep command 105
first level directory, restoring from tape 64
force console command 33, 35, 137, 298
forcing password changes 34
free blocks 72
free list
  add space from Dead queue 130
  high water mark 43
  introduction 43
  low water mark 43
  rebuild 133
  size 124
free space 72
FTS servers 109
  checking communication 110
ftsdebug, format 109
ftsidx.exe 109
ftssch.exe 109
fullform job list command 169
function key console control command 117

## G

grep 49
group security
  controlling system access 39
  designing 39
groups
  managing from console 329
grpcheck command 137
gtraits command, syntax 137

## H

help console command 138
HEOL dialog command 172
history command
  bottom 116
  list 117
  print 118

top 118
up 119
hogs command 45, 138

## I

Ierrs, input errors 109
ignore command 165
ignore-del job list command 165
increasing database space 46
iNEWS
checking server communication 110
integration 12
Linux commands available 115
overview 11
input errors (Ierrs) 109

## J

job list command
at 163
bscan 163, 164, 167
distribution 164
dup 164
every 164
fullform 169
ignore 165
ignore-del 165
local 165
move 165
number 165
on 166
open 166
order 166
put 167
quiet 167
remote 168
remove 168
replace 168
scan 168
send-del 169
source 169
validate 169

## L

Last Login option 36
line formatting, heol command 174
Linus commands 115
list console control command 117
list d command
syntax 142–143
list g command
syntax 143
list q command 143
syntax 143
list s command 144
list u command 96, 144, 304
syntax 144
list u-v command, user traits 296
listing all computers 121
listing contents of a tape 59
listing last password change 301
listing password change by date 302–304
listing users logged in 37
loading tapes 56
local job list command 165
locked blocks
checking for 105
locked stories, removing 127
logclose console control command 117
logging in
users unable to 96
logins, recording 38
logopen console control command 117
logout command 118, 146
low on space message 43
low water mark, free list 43

## M

mailboxes
  assigning to queues 327
  listing all queues (at console) 327
maintaining database storage 48–53
makeccutab console command 147
makeshift command 242
maketab console command 149
making tape backups 156
map dialog command 173
mapin dialog command 173
mapout dialog command 173
master computer, putting back online 92
mcs dictionary
  driver errors 246, 247
  mct driver errors 246
  shared error messages 246
  status reports 247
message dialog command 21, 174
messages, broadcast 120
minus four option 102
modem console control command 118
modems, resetting for a wire 102
MODIFY-DEV field 30
monitoring free space 42–44
monitoring the free list 43
move job list command 165
msgclean console command 150
msgmailalert 280

## N

netstat -i 84, 108
network failure 107
network services
  adding 23
  adding Linux 25
number job list command 165
number of blocks, list 123
nxserver 291

## O

Oerrs, output errors 108
offline command 150
on job list command 166
online command 150
open job list command 166
order job list command 166
order lock, unbusying 100
otod console command 151
out-of-space condition 44
output errors (Oerrs) 108

## P

pass dialog command 174
passwords
  checking 304
  checking for users without 31
  forcing change 34
  listing change by date 302–304
  listing last change 34, 301
  required minimum length 30
  restrictions 35
  security procedures 30
  setting and changing for users 31
  status 31
pause command 174
PCU
  back panel 253
  ccumsgs dictionary 213
  ccureset command 152
  introduction 250
peripheral controller unit *See* PCU
picolor 282
ping, testing connectivity 107
piping with fgrep 105

polarity 103
policies for backups 56
poll job list command 167
power failure 106
print command 118, 152
PRINTER display, console 117
printer profile
  corresponding to printer ports 104
printing
  deleting runaway print jobs 104
  system printer problems 103
problems with wires 101
procedures
  back up site files to tape 76
  change presenter instructions text color 282
  define key for delete_notify 290
  define key used to advance timing bar 284
  enable destination order 278
  enable message mail alerts 281
  enable synchronized timing 289
  exit console program 117
  hide groups from e-mail lists 279
  recover from disconnect 85
  search user's last login 36
  set up system service 23
  start or restart queue sorting 323
processes status 105
ps command 105
purge interval
  database trait 323
  defined 42
  introduction 42
  setting 323
purging stories 42
put job list command 167

## Q

Q.SEEK 223
QIC tapes 56
queues
  backing up 60
  creating a dialog 20
  dictionary 223
  sort 131
quiet job list command 167

## R

reclaiming Dead queue space 46
reconnect command 153
recording logins 38
recovering a failed system 86
recovering a story
  by word 69
  maximum 71
red data, polarity 103
references in a story 44
relative date, defined 302
remote job list command 168
remove job list command 168
removing
  locks 122
  stories 126
rename command 154
replace job list command 168
reset console control command 118
resetting a wire's modem 102
restart console command 154
restoring dictionary defaults 201
restoring from tape
  checking space available 72
  first level directory 64
  list date of each backup 67
  listing tape backup dates 66
  listing tape contents 65
rotating tape dumps 56
rr kb su m SOEKCVTH sc 145
runaway print jobs
  eliminating 104

rundown queue
display 328
setting display lines 328
RXDEBUG= 293
RXDEBUGFILE 293
RXSITELISTSZ= 293

## S

scan job list command 168
scanning queues 122
searching
by word and date 70
for stories, word and day 70
for stories, word and month 71
specifying number of stories 71
searching a tape 68
by date 70
searchtape command 68
searchtape console command 155
security
groups 39
protecting SYSTEM directory
from console 331
security modem 30
security procedures ??–31
MODIFY-DEV field 30
passwords 30
security modem 30
superuser status 30
send console command 155
send-del job list command 169
server 291
servers
checking communication 110
FTS utility programs 109
service line
adding 23
service table
adding a network service 23
adding a service 24
line format 25
parameters 25
services
adding 23
adding dialogs 20
attaching a dialog 25
character mapping 173
defined 19
installing 23
network 23
parts of 23
restricting access 25
setting
a purge interval 323
user passwords 31
showtimingbar 284
shutdown console command 156
site files, backing up 76
/site/dict/dial 236
/site/dict/shift 240
sitedump command 76, 156
siterestore console command 156
softdump command 157
softrestore console command 157
sortfield database trait 321
sorting stories 131
source job list command 169
space
blocks 43
free list 43
hogs command 45
increasing in the database 46
reclaiming 46
usage 44
SRPlo-LIsUGQSXWFi, explained 312
startup after power failure 122
startup command 158
status of passwords 31
stop console command 159

stop dialog command 175
storage units 43
stories
  cannot read or write 98
  error checking 126
  export in case of disconnect 81
  how the system copies 44
  purging 42
  references 44
  removes old versions 126
  retrieve backup from tape 128
  sort 131
  storing copies 44
  unbusying 99
superuser
  console command 159
synctoserver 288
system
  adding services 23
  list all computers 153
  security 29–40
system access
  group security 39
system administrator
  basics 13
  customization 15
  database tasks 14
  security 15
  storage maintenance 15
  troubleshooting 17
  user tasks 14
SYSTEM directory
  hiding 331
  restricting 331
system failure, recovering 86
system servers, FTS 109
SYSTEM.DIALOG 27
SYSTEM.DIALOG directory 20
SYSTEM.PRINTERS 103
SYSTEM.SEARCHTAPE queue 68
SYSTEM.SERVICE 23

## T

table formatting, heol command 174
tape contents, listing 65
tapes
  backup 123
  listing contents 59
  loading 56
  rotating dumps 56
test communication 152
timer
  dialog command 175
  setting 175
TK-50 tapes 56
top console control command 118
tracking database space 44
tracking user login activity 36
trait listing
  inverted 309
  read-only 309
  sequentially 309
type a string 175
type dialog command 175

## U

unbusying
  a story 99
  Dead queue 100
  edit lock 100
  order lock 100
Unity for News 10
up console control command 119
user activity, tracking 35–39
user passwords
  setting and changing 31
user traits

list u-v command 296
password restrictions 35
summary of 304
users
cannot read or write stories 98
last password change 34
listing log ins 37
unable to log in 96
Users Without Passwords option 33
utraits console command 160
changing a user's password 34


## V


validate job list command 169
verbose command
getting detailed information 310
verifying a backup tape 59
version console command 161
view console control command 119


## W


W_LOGTYPES value 38
wait dialog command 176
wholockedit command 162
wire ports 101
wire problems 101
wiredump console command 162
workstation problems 96
write-protect tab on tapes 56


## X


x console control command 119
xon/xoff flow control 103


## Z


zoom console control command 120